

Kohei Arai *Editor*

Intelligent Computing

Proceedings of the 2021 Computing Conference, Volume 2



Springer

Lecture Notes in Networks and Systems

Volume 284

Series Editor

Janusz Kacprzyk, Systems Research Institute, Polish Academy of Sciences,
Warsaw, Poland

Advisory Editors

Fernando Gomide, Department of Computer Engineering and Automation—DCA,
School of Electrical and Computer Engineering—FEEC, University of Campinas—
UNICAMP, São Paulo, Brazil

Okyay Kaynak, Department of Electrical and Electronic Engineering,
Bogazici University, Istanbul, Turkey

Derong Liu, Department of Electrical and Computer Engineering, University
of Illinois at Chicago, Chicago, USA; Institute of Automation, Chinese Academy
of Sciences, Beijing, China

Witold Pedrycz, Department of Electrical and Computer Engineering,
University of Alberta, Alberta, Canada; Systems Research Institute,
Polish Academy of Sciences, Warsaw, Poland

Marios M. Polycarpou, Department of Electrical and Computer Engineering,
KIOS Research Center for Intelligent Systems and Networks, University of Cyprus,
Nicosia, Cyprus

Imre J. Rudas, Óbuda University, Budapest, Hungary

Jun Wang, Department of Computer Science, City University of Hong Kong,
Kowloon, Hong Kong

The series “Lecture Notes in Networks and Systems” publishes the latest developments in Networks and Systems—quickly, informally and with high quality. Original research reported in proceedings and post-proceedings represents the core of LNNS.

Volumes published in LNNS embrace all aspects and subfields of, as well as new challenges in, Networks and Systems.

The series contains proceedings and edited volumes in systems and networks, spanning the areas of Cyber-Physical Systems, Autonomous Systems, Sensor Networks, Control Systems, Energy Systems, Automotive Systems, Biological Systems, Vehicular Networking and Connected Vehicles, Aerospace Systems, Automation, Manufacturing, Smart Grids, Nonlinear Systems, Power Systems, Robotics, Social Systems, Economic Systems and other. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution and exposure which enable both a wide and rapid dissemination of research output.

The series covers the theory, applications, and perspectives on the state of the art and future developments relevant to systems and networks, decision making, control, complex processes and related areas, as embedded in the fields of interdisciplinary and applied sciences, engineering, computer science, physics, economics, social, and life sciences, as well as the paradigms and methodologies behind them.

Indexed by SCOPUS, INSPEC, WTI Frankfurt eG, zbMATH, SCImago.

All books published in the series are submitted for consideration in Web of Science.

More information about this series at <http://www.springer.com/series/15179>

Kohei Arai
Editor

Intelligent Computing

Proceedings of the 2021 Computing
Conference, Volume 2



Springer

Editor

Kohei Arai

Faculty of Science and Engineering

Saga University

Saga, Japan

ISSN 2367-3370

ISSN 2367-3389 (electronic)

Lecture Notes in Networks and Systems

ISBN 978-3-030-80125-0

ISBN 978-3-030-80126-7 (eBook)

<https://doi.org/10.1007/978-3-030-80126-7>

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Editor's Preface

It is a great privilege for us to present the proceedings of the Computing Conference 2021, held virtually on July 15 and 16, 2021.

The conference is held every year to make it an ideal platform for researchers to share views, experiences and information with their peers working all around the world. This is done by offering plenty of networking opportunities to meet and interact with the world-leading scientists, engineers and researchers as well as industrial partners in all aspects of computer science and its applications.

The main conference brings a strong program of papers, posters, videos, all in single-track sessions and invited talks to stimulate significant contemplation and discussions. These talks were also anticipated to pique the interest of the entire computing audience by their thought-provoking claims which were streamed live during the conferences. Moreover, all authors had very professionally presented their research papers which were viewed by a large international audience online.

The proceedings for this edition consist of 235 chapters selected out of a total of 638 submissions from 50+ countries. All submissions underwent a double-blind peer-review process. The published proceedings has been divided into three volumes covering a wide range of conference topics, such as technology trends, computing, intelligent systems, machine vision, security, communication, electronics and e-learning to name a few.

Deep appreciation goes to the keynote speakers for sharing their knowledge and expertise with us and to all the authors who have spent the time and effort to contribute significantly to this conference. We are also indebted to the organizing committee for their great efforts in ensuring the successful implementation of the conference. In particular, we would like to thank the technical committee for their constructive and enlightening reviews on the manuscripts in the limited timescale.

We hope that all the participants and the interested readers benefit scientifically from this book and find it stimulating in the process.

Hope to see you in 2022, in our next Computing Conference, with the same amplitude, focus and determination.

Kohei Arai

Contents

Balanced Weighted Label Propagation	1
Matin Pirouz	
Analysis of the Vegetation Index Dynamics on the Base of Fuzzy Time Series Model	13
Elchin Aliyev, Ramin Rzayev, and Fuad Salmanov	
Application of Data-Driven Fault Diagnosis Design Techniques to a Wind Turbine Test-Rig	23
Silvio Simani, Saverio Farsoni, and Paolo Castaldi	
Mathematical Model to Support Decision-Making to Ensure the Efficiency and Stability of Economic Development of the Republic of Kazakhstan	39
Askar Boranbayev, Seilkhan Boranbayev, Malik Baimukhamedov, and Askar Nurbekov	
Automated Data Processing of Bank Statements for Cash Balance Forecasting	49
Vlad-Marius Griguta, Luciano Gerber, Helen Slater-Petty, Keeley Crocket, and John Fry	
Top of the Pops: A Novel Whitelist Generation Scheme for Data Exfiltration Detection	65
Michael Cheng Yi Cho, Yuan-Hsiang Su, Hsiu-Chuan Huang, and Yu-Lung Tsai	
Analyzing Co-occurrence Networks of Emojis on Twitter	80
Hasan Alsaif, Phil Roesch, and Salem Othman	
Development of an Oceanographic Databank Based on Ontological Interactive Documents	97
Oleksandr Stryzhak, Vitalii Prykhodniuk, Maryna Popova, Maksym Nadutenko, Svitlana Haiko, and Roman Chepkov	

Analyzing Societal Bias of California Police Stops Through Lens of Data Science	115
Sukanya Manna and Sara Bunyard	
Automated Metadata Harmonization Using Entity Resolution and Contextual Embedding	129
Kunal Sawarkar and Meenakshi Kodati	
Data Segmentation via t-SNE, DBSCAN, and Random Forest	139
Timothy DeLise	
GeoTree: A Data Structure for Constant Time Geospatial Search Enabling a Real-Time Property Index	152
Robert Miller and Phil Maguire	
Prediction Interval of Future Waiting Times of the Two-Parameter Exponential Distribution Under Multiply Type II Censoring	166
Shu-Fei Wu	
Predictive Models as Early Warning Systems: A Bayesian Classification Model to Identify At-Risk Students of Programming	174
Ashok Kumar Veerasamy, Mikko-Jussi Laakso, Daryl D'Souza, and Tapio Salakoski	
Youden's J and the Bi Error Method	196
MaryLena Bleile	
A New Proposal of Parametric Similarity Measures with Application in Decision Making	210
Luca Anzilli and Silvio Giove	
The Challenges of Using Big Data in the Consumer Credit Sector	221
Kirill Romanyuk	
New Engine to Promote Big Data Industry Upgrade	232
Jing He, Chuyi Wang, and Haonan Chen	
Using Correlation and Network Analysis for Researching Intellectual Competence	249
Sipovskaya Yana Ivanovna	
A Disease Similarity Technique Using Biological Process Functional Annotations	261
Luis David Licea Torres and Hisham Al-Mubaid	
Impact of Types of Change on Software Defect Prediction	273
Atakan Erdem	
Analyzing Music Genre Popularity	284
Jose Fossi, Adam Dzwonkowski, and Salem Othman	

Result Prediction Using Data Mining	295
Hasan Sarwar, Dipannoy Das Gupta, Sanzida Mojib Luna, Nusrat Jahan Suhi, and Marzouka Tasnim	
A Systematic Literature Review on Big Data Extraction, Transformation and Loading (ETL).....	308
Joshua C. Nwokeji and Richard Matovu	
Accelerating Road Sign Ground Truth Construction with Knowledge Graph and Machine Learning	325
Ji Eun Kim, Cory Henson, Kevin Huang, Tuan A. Tran, and Wan-Yi Lin	
Adjusted Bare Bones Fireworks Algorithm to Guard Orthogonal Polygons	341
Adis Alihodzic, Damir Hasanspahic, Fikret Cunjalo, and Haris Smajlovic	
Unsupervised Machine Learning-Based Elephant and Mice Flow Identification	357
Muna Al-Saadi, Asiya Khan, Vasilios Kelefouras, David J. Walker, and Bushra Al-Saadi	
Automated Generation of Zigzag Carbon Nanotube Models Containing Haeckelite Defects	371
M. Leonor Contreras, Ignacio Villarroel, and Roberto Rozas	
Artificial Intelligence Against Climate Change	378
Leila Scola	
Construction Site Layout Planning Using Multiple-Level Simulated Annealing	398
Hui Jiang and YaBo Miao	
Epistocracy Algorithm: A Novel Hyper-heuristic Optimization Strategy for Solving Complex Optimization Problems	408
Seyed Ziae Mousavi Mojab, Seyedmohammad Shams, Hamid Soltanian-Zadeh, and Farshad Fotouhi	
An Approach for Non-deterministic and Automatic Detection of Learning Styles with Deep Belief Net	427
Maxwell Ndognkon Manga and Marcel Fouada Ndjodo	
Using Machine Learning to Identify Methods Violating Immutability	453
Tamás Borbély, Árpád János Wild, Balázs Pintér, and Tibor Gregoris	
Predicting the Product Life Cycle of Songs on the Radio.....	463
O. F. Grooss, C. N. Holm, and R. A. Alphinias	

Hierarchical Roofline Performance Analysis for Deep Learning Applications	473
Charlene Yang, Yunsong Wang, Thorsten Kurth, Steven Farrell, and Samuel Williams	
Data Augmentation for Short-Term Time Series Prediction with Deep Learning	492
Anibal Flores, Hugo Tito-Chura, and Honorio Apaza-Alanoca	
On the Use of a Sequential Deep Learning Scheme for Financial Fraud Detection	507
Georgios Zioviris, Kostas Kolomvatsos, and George Stamoulis	
Evolutionary Computation Approach for Spatial Workload Balancing.....	524
Ahmed Abubahia, Mohamed Bader-El-Den, and Ella Haig	
Depth Self-optimized Learning Toward Data Science	543
Ziqi Zhang	
Predicting Resource Usage in Edge Computing Infrastructures with CNN and a Hybrid Bayesian Particle Swarm Hyper-parameter Optimization Model	562
John Violas, Tita Pagoulatou, Stylianos Tsanakas, Konstantinos Tserpes, and Theodora Varvarigou	
Approaching Deep Convolutional Neural Network for Biometric Recognition Based on Fingerprint Database	581
Md. Saiful Islam, Tanhim Islam, and Mahady Hasan	
Optimizing the Neural Architecture of Reinforcement Learning Agents	591
N. Mazyavkina, S. Moustafa, I. Trofimov, and E. Burnaev	
Automatic Ensemble of Deep Learning Using KNN and GA Approaches	607
Ben Zagagy, Maya Herman, and Ofer Levi	
A Deep Learning Model for Data Synopses Management in Pervasive Computing Applications	619
Panagiotis Fountas, Kostas Kolomvatsos, and Christos Anagnostopoulos	
DeepObfuscation: Source Code Obfuscation through Sequence-to-Sequence Networks	637
Siddhartha Datta	
Deep-Reinforcement-Learning-Based Scheduling with Contiguous Resource Allocation for Next-Generation Wireless Systems	648
Shu Sun and Xiaofeng Li	

A Novel Model for Enhancing Fact-Checking	661
Fatima T. AlKhawaldeh, Tommy Yuan, and Dimitar Kazakov	
Investigating Learning in Deep Neural Networks Using Layer-Wise Weight Change.....	678
Ayush Manish Agrawal, Atharva Tendle, Harshvardhan Sikka, Sahib Singh, and Amr Kayid	
Deep Reinforcement Learning for Task Planning of Virtual Characters	694
Caio Souza and Luiz Velhor	
Accelerating Deep Convolutional Neural on GPGPU	712
Dominik Źurek, Marcin Pietroń, and Kazimierz Wiatr	
Enduring Questions, Innovative Technologies: Educational Theories Interface with AI	725
Rosemary Papa and Karen Moran Jackson	
DRAM-Based Processor for Deep Neural Networks Without SRAM Cache	743
Eugene Tam, Shenfei Jiang, Paul Duan, Shawn Meng, Yue Pan, Cayden Huang, Yi Han, Jacke Xie, Yuanjun Cui, Jinsong Yu, and Minggui Lu	
Study of Residual Networks for Image Recognition	754
Mohammad Sadegh Ebrahimi and Hossein Karkeh Abadi	
A Systematic Review of Educational Data Mining	764
FangYao Xu, ZhiQiang Li, JiaQi Yue, and ShaoJie Qu	
Image Classification with A-MnasNet and R-MnasNet on NXP Bluebox 2.0	781
Prasham Shah and Mohamed El-Sharkawy	
CreditX: A Decentralized and Secure Credit Platform for Higher Educational Institutes Based on Blockchain Technology	793
Romesh Liyanage, D. P. P. Jayasinghe, K. T. Uvindu Sanjana, H. B. D. R. Pearson, Disni Sriyaratna, and Kavinga Abeywardena	
An Architecture for Blockchain-Based Cloud Banking	805
Thuat Do	
A Robust and Efficient Micropayment Infrastructure Using Blockchain for e-Commerce	825
Soumaya Bel Hadj Youssef and Noureddine Boudriga	
Committee Selection in DAG Distributed Ledgers and Applications	840
Bartosz Kuśmierz, Sebastian Müller, and Angelo Capossele	

An Exploration of Blockchain in Social Networking Applications	858
Rituparna Bhattacharya, Martin White, and Natalia Beloff	
Real and Virtual Token Economy Applied to Games: A Comparative Study Between Cryptocurrencies	869
Isabela Ruiz Roque da Silva and Nizam Omar	
Blockchain Smart Contracts Static Analysis for Software Assurance	881
Suzanna Schmeelk, Bryan Rosado, and Paul E. Black	
Promize - Blockchain and Self Sovereign Identity Empowered Mobile ATM Platform	891
Eranga Bandara, Xueping Liang, Peter Foytik, Sachin Shetty, Nalin Ranasinghe, Kasun De Zoysa, and Wee Keong Ng	
Investigating the Robustness and Generalizability of Deep Reinforcement Learning Based Optimal Trade Execution Systems	912
Siyu Lin and Peter A. Beling	
On Fairness in Voting Consensus Protocols	927
Sebastian Müller, Andreas Penzkofer, Darcy Camargo, and Olivia Saa	
Dynamic Urban Planning: An Agent-Based Model Coupling Mobility Mode and Housing Choice. Use Case Kendall Square	940
Mireia Yurrita, Arnaud Grignard, Luis Alonso, Yan Zhang, Cristian Ignacio Jara-Figueroa, Markus Elkatcha, and Kent Larson	
Reasoning in the Presence of Silence in Testimonies: A Logical Approach	952
Alfonso Garcés-Báez and Aurelio López-López	
A Genetic Algorithm Based Approach for Satellite Autonomy	967
Sidhdharth Sikka and Harshvardhan Sikka	
Communicating Digital Evolutionary Machines	976
Istvan Elek, Zoltan Blazsik, Tamas Heger, Daniel Lenger, and Daniel Sindely	
Analysis of the MFC Singularities of Speech Signals Using Big Data Methods	987
Ruslan V. Skuratovskii and Volodymyr Osadchy	
A Smart City Hub Based on 5G to Revitalize Assets of the Electrical Infrastructure	1010
Santiago Gil, Germán D. Zapata-Madrigal, and Rodolfo García Sierra	
LoRa RSSI Based Outdoor Localization in an Urban Area Using Random Neural Networks	1032
Winfred Ingabire, Hadi Larjani, and Ryan M. Gibson	

A Deep Convolutional Neural Network Approach for Plant Leaf Segmentation and Disease Classification in Smart Agriculture	1044
Ilias Masmoudi and Rachid Lghoul	
Medium Resolution Satellite Image Classification System for Land Cover Mapping in Nigeria: A Multi-phase Deep Learning Approach	1056
Nzurumike L. Obianuju, Nwojo Agwu, and Onyenwe Ikechukwu	
Analysis of Prediction and Clustering of Electricity Consumption in the Province of Imbabura-Ecuador for the Planning of Energy Resources	1073
Jhonatan F. Rosero-Garcia, Edilberto A. Llanes-Cedeño, Ricardo P. Arciniega-Rocha, and Jesús López-Villada	
Street Owl. A Mobile App to Reduce Car Accidents	1085
Mohammad Jabrah, Ahmed Bankher, and Bahjat Fakieh	
Low-Cost Digital Twin Framework for 3D Modeling of Homogenous Urban Zones	1106
Emad Felemban, Abdur Rahman Muhammad Abdul Majid, Faizan Ur Rehman, and Ahmed Lbath	
VR in Heritage Documentation: Using Microsimulation Modelling	1115
Wael A. Abdelhameed	
MQTT Based Power Consumption Monitoring with Usage Pattern Visualization Using Uniform Manifold Approximation and Projection for Smart Buildings	1124
Ray Mart M. Montesclaros, John Emmanuel B. Cruz, Raymark C. Parocha, and Erees Queen B. Macabebé	
Deepened Development of Industrial Structure Optimization and Industrial Integration of China's Digital Music Under 5G Network Technology	1141
Li Eryong and Li Yukun	
Optimal Solution of Transportation Problem with Effective Approach Mount Order Method: An Operational Research Tool	1151
Mohammad Rashid Hussain, Ayman Qahmash, Salem Alelyani, and Mohammed Saleh Alsaqer	
Analysis of Improved User Experience When Using AR in Public Spaces	1169
Vladimir Barros, Eduardo Oliveira, and Luiz Araújo	
Autonomous Vehicle Decision Making and Urban Infrastructure Optimization	1190
George Mudrak and Sudhanshu Kumar Semwal	

Prediction of Road Congestion Through Application of Neural Networks and Correlative Algorithm to V2V Communication (NN-CA-V2V)	1203
Mahmoud Zaki Iskandarani	
Urban Planning to Prevent Pandemics: Urban Design Implications of BiocyberSecurity (BCS)	1222
Lucas Potter, Ernestine Powell, Orlando Ayala, and Xavier-Lewis Palmer	
Smart Helmet: An Experimental Helmet Security Add-On	1236
David Sales, Paula Prata, and Paulo Fazendeiro	
Author Index	1251



Balanced Weighted Label Propagation

Matin Pirouz^(✉)

California State University, Fresno, CA 93740, USA
mpirouz@ieee.org

Abstract. One key methodology to understand the structure of complex networks is through community detection and analysis. Such information is used to find relationships and hidden structures in social communities. Existing community detection algorithms are either computationally expensive in large-scale real-world networks or require specific information such as the number and size of communities. Another problem with the existing benchmark algorithms is the resolution problem, i.e. as the data grows larger or more complex (with a higher chance of having outliers), the identified community structure loses quality. In this paper, we introduce a multi-stage novel edge-influenced label propagation algorithm that uses the network structure to create values for the edges. Initially, edges are used to create the flow of the community structure. Next, every node is initialized with a unique label and at every step, each node adopts the label of the neighbor with the lowest edge weight. Finally, all nodes converge and construct the community structures. Through an iterative process, densely connected groups of nodes form a consensus on a unique label to form communities. The proposed method, Balanced Weighted Label Propagation, is tested on both synthetic and real-world benchmark datasets with known community structures.

Keywords: Social network analysis · Normalized Mutual Information · Adjusted Rand Index · Resolution limit

1 Introduction

Networks are groups of nodes inter-connected with edges. For example in social media networks, nodes represent people and edges represent the relationship among people [10]. There exist some subgroups with more internal edges than external edges, commonly identified as communities. One of the main tasks in studying complex networks is to find these structures. Finding communities can result in finding meaningful information, and is particularly important in social networks [15] and recommender systems [3, 19].

Several community detection methods exist. Modularity, a quality-function introduced by Newman, has become one of the major algorithms for which there have been many optimization methods. On the other hand, the resolution limits [12] problem found by Fortunato and Barthélemy in [1] is a key limitation

of modularity. A metric called resolution value has been used to solving the problem, where the higher the value, the more communities are found [7].

Another widely known approach is Label Propagation developed by Raghavan et al. [12]. Label Propagation is a near linear approach, which is known as the coloring method as it is based on the flow of uniquely assigned labels in the network. The problem with this method is that the core method is based on random distribution; therefore, many different scenarios of answer could be resulted. Several enhancements have been introduced over the years to solve the problem and are further studied in the related work section.

In this paper, a new node influence algorithm based on Label Propagation is proposed to solve the randomness and resolution limit problem in community detection. Balanced Weighted Label Propagation (BWLP) algorithm has a time complexity linear to the number of edges. Our approach uses the influence of nodes on one another to find an edge value for any pair of nodes in any complex graph. The neighboring relationship between two nodes is used as a metric to define the closeness of the pair. We develop a convergence function which defines when to stop exchanging labels. Finally, when the distribution of labels concludes, all the nodes that share the same label find them selves in the same community.

1.1 Contributions

By computing the edge weight based on dissimilarity between nodes, BWLP achieves the following:

- Intuitive Community Detection: Contrary to solely relying on neighboring labels count in which is not the most efficient way, BWLP uses weighted flow to find community structure.
- Hidden Community Detection: In large-scale networks, hidden and small communities can be neglected due to various factors (as explained in Sect. 3.1). Relying on the local topology-driven weighted edges, BWLP enables the discovery of small communities.
- Scalability: BWLP algorithm is superlinear to the number of nodes, which results in being scalable for large networks. BWLP only iterates over edges around every node and there is no running over the same node twice. As a result, BWLP achieves a time complexity of $O(|n|)$. Given this property, BWLP lends itself to handling large real-world networks in a relatively short time.
- Addressing the resolution limit ring problem: Using edge weights and structure flow, we address the resolution problem.

2 Related Work

Community detection algorithms vary based on their approach and strengths/limitations. CPM [8] searches the graph for all cliques of maximum

size as the first phase. After the maximum cliques have been found, each clique is enumerated and a clique adjacency matrix is constructed. A parameter K is set and any clique with size of at least K is considered to be a “part” of a larger module. Girvan and Newman method [2] creates communities by removing the highest edge betweenness and re-calculating influenced edges value again. At the end, it provides a dendrogram with nodes as the leaves. The resolution of communities in all the optimization method with a fitness function is dependent on the resolution value chosen for the given function. Lancichinetti et al. [4] choose a random node and expands a community from there. Similar to modularity, resolution value defines the size and quality of the detected communities depends. In EAGLE [14] and GCE [6] algorithms, maximal cliques in the network replaces the random node used in [4].

Zhang et al.’s spectral method [22] is used to lower the dimensions of the graph. The fuzzy c-mean algorithm is then used for clustering. Psorakis et al. [11] introduced a novel method utilizing Bayesian non-negative matrix factorization (NMF). The limitations of this method include the need to know the number of communities and the use of matrices which results in an inefficient time complexity and increases the space complexity. Zhang et al. [21] used a stochastic block model to find an unequal size community structure where the number of nodes inside community should be equal when using stochastic block models.

Label propagation methods [18] propagate the label of the most popular nodes throughout the graph to find the structure of communities. Particle Swarm Optimization (PSO) [23] introduced a new label propagation method mixed with modularity density. The detected communities have good resolution but the algorithm has a higher time complexity than other label propagation algorithms. SLPA introduced a dynamic algorithm to find overlapping community structure. Li et al. [17] introduced an optimized method called “Stepping LPA-S”, where similarity for propagating labels is used. In addition, a stepping framework is used to divide networks. Then, an evaluation function is used to select the final unique partition.

3 Proposed Approach

This section develops the main idea behind the proposed multi-stage edge weighting algorithm. In the initialization layer, all the links get their own labels. Assume a pair of nodes with unique sets of common and exclusive neighbors. Suppose X and Y are neighbors, then C is the set of their common neighbors and E is the set of exclusive neighbors between the pair. In this study, we use sets C and E to find the dissimilarity between nodes X and Y . Function 2 finds a dissimilarity value between any pair of nodes in the graph based on their structural formats. The calculated dissimilarity value is assigned to the edge between two nodes as their edge weight. Algorithm 1 shows the next step which is sorting all nodes based on their dissimilarity ratio [9].

First, every node in the graph gets their node number as their label. This way, every node is assigned to its own community. Later, swapping labels

begins as shown in Algorithm 2. In every iteration, vertices will change membership to the community of the most similar vertex around them. To choose which community the node X belongs to, X looks around among its neighbors $\mathcal{N}_X = \{n_1, n_2, n_3, \dots, n_i\}$ and joins the community of the neighbor which has the lowest dissimilarity to itself among all its neighbors using set $\mathcal{W}_{\mathcal{N}_X} = \{w_1, w_2, w_3, \dots, w_i\}$. This way, X picks the label of its most influential neighbor. In case of a tie of edge weights among the neighbors of X , the algorithm chooses labels with the least number of exclusive nodes.

In parallel with the label exchange, Convergence Algorithm 3 takes place. Convergence is the process of every node checking whether they are in the correct community or not. Every node holds a binary value which decides if the node should continue searching for the community it belongs to or it has the right label. This value is initially set to false for every node. If the majority of set $\mathcal{N}_X = \{n_1, n_2, n_3, \dots, n_i\}$ for X share the same label of X , the value changes to true and X stops searching. On the other hand, if the majority of neighbors do not agree with X 's label, X look around the neighbors of which has the lowest edge weight to it. This continues until the neighbors agree on X 's label. By the end of each iteration, all the nodes holding the same label count as one community.

Label propagation is an iterative clustering procedure, in which at each iteration the label (cluster) of a node is updated based on a function of graph topology. In this algorithm, the edge weighted function is based on the shared neighbors of the nodes in question. More formally, we let $G(V, E)$ be an unweighted graph with vertex set V and edge set E . For a node $x \in V$, let $\mathcal{N}(X) := \{y \mid (x, y) \in E\}$, let $D(x)$ be the degree of x , and define the function $F : V \times V \rightarrow \mathbb{R}$ as

$$F(x, y) = \frac{-2|\mathcal{N}(x) \cap \mathcal{N}(y)| + |\mathcal{N}(x)| + |\mathcal{N}(y)|}{D(x) * D(y)}. \quad (1)$$

If $G(V, E)$ is a simple graph, then $D(x) = |\mathcal{N}(x)|$ and the function can be written as

$$F(x, y) = \frac{-2|\mathcal{N}(x) \cap \mathcal{N}(y)|}{|\mathcal{N}(x)| * |\mathcal{N}(y)|} + \frac{1}{\mathcal{N}(x)} + \frac{1}{\mathcal{N}(y)}. \quad (2)$$

The label propagation algorithm has two main procedures, an initialization procedure, and an iterative procedure. The initialization process begins by assigning each edge $(x, y) \in E$ a weight given by $F(x, y)$ and each node $x \in V$ a unique label denoted by $label.x$. At each step in the iterative procedure, each node is considered and assigns a new label $label.y$ where $y \in V$ satisfies the following:

$$\min_{y \in \mathcal{N}(x)} F(x, y). \quad (3)$$

It is known that modularity maximization suffers from the resolution problem. Namely, it has a tendency to cluster the k-cliques together in the ring of cliques network. In the following lemma, we show that our algorithm will never cluster the k-cliques together if $k > 2$.

Algorithm 1: Weighted Label Propagation

```

1 Input:  $G$  network graph,  $w_{min}$ 
2 Output: network communities
3 for  $n \in G$  do
4    $n_{label} = n$ 
5   for  $i \in \{Neighbors\ of\ n\}$  do
6      $[N_w] = \frac{-2|\mathcal{N}(n) \cap \mathcal{N}(i)| + |\mathcal{N}(n)| + |\mathcal{N}(i)|}{D(n)*D(i)}$ 
7   end
8   for  $j \in \{SNL\}$  do
9     if  $\min\{N_w\} < SNL[j]$  then
10       $SNL[j] \leftarrow insert\ n$ 
11    end
12  end
13 end

```

Algorithm 2: Map Traversing

```

1 Input:  $G$  network graph,  $w_{min}$ ,  $SNL$ 
2 Output: network communities
3 while  $N_c \neq 0$  do
4   for  $n \in SNL$  do
5     for  $i \in \{Neighbors\ of\ n\}$  do
6       if  $n_w^i < Chosen$  then
7         chosen= $label_{n_w^i}$ 
8       end
9       for  $i \in \{Neighbors\ of\ n\}$  do
10         HAL= Find highest occurrence label
11       end
12     end
13     if  $HAL == Chosen$  then
14       Flag  $n$  as converged
15     end
16   else
17     call  $Convergence(n, G)$ ;
18   end
19 end
20 end

```

Lemma 1. *If the given algorithm converges on the ring of cliques network where each clique has the size $k > 2$, then no node will be clustered with nodes outside of its clique.*

Proof. This lemma is proven if we can show that no node has the incentive to cluster with a node outside of its clique. To this end, let $x \in V$ and define $C(x)$ to be the set of nodes that share a clique with x . Let y be a node in the set

Algorithm 3: Convergence

```

1 Input:  $G$  network graph,  $n$ , Chosen
2 for  $i \in \{Neighbors\}$  do
3   Chosen = label of neighbor with most similarity
4   m = neighbor with lowest edge value
5   for  $j \in \{Neighbors\}$  do
6     HAL= Find highest occurrence label
7   end
8   if HAL == Chosen then
9     Flag n as converged
10  end
11 else
12   call Convergence(m, G);
13 end
14 end

```

$\mathcal{N}(x) - C(x)$ and notice that by the construction of the ring of cliques we have

$$|\mathcal{N}(x) \cap \mathcal{N}(y)| = 0. \quad (4)$$

depending on how we construct the ring of cliques and the choice of x and y $F(x, y)$'s value will vary; however

$$\min F(x, y) = \frac{2}{k+1}. \quad (5)$$

Let z be any node in $C(x)$. Since z belongs to the same clique as x we have

$$|\mathcal{N}(x) \cap \mathcal{N}(z)| = k - 2. \quad (6)$$

Again, depending on how we construct the ring of cliques and the choice of x and z $F(x, z)$'s value will vary; however

$$\max F(x, z) = \frac{-2(k-2)}{(k+2)(k)} + \frac{1}{k+2} + \frac{1}{k}. \quad (7)$$

Finally, we note that $\max F(x, z) < \min F(x, y)$ as long as $k > \frac{1}{2}(1 + \sqrt{13})$ since we assumed that k is an integer that is greater x will never cluster with a node outside of its clique.

Table 1. Real world unweighted datasets [20]

Datasets	Num. nodes	Num. edges	Connected component	CC	Avg. degree	Network diameter
polbooks	105	441	1	0.488	8.4	7
Football	115	613	1	0.403	10.661	4
Dolphin social network	62	159	1	0.303	5.129	8
Karate	34	78	1	0.285	4.588	3

Table 2. Synthetic unweighted datasets created with LFR benchmark graphs

Datasets	Num. nodes	Num. edges	Connected component	CC	Avg. degree	Network diameter
SM1	1000	4288	3	0.477	4.28	18
SM2	1000	4334	4	0.526	4.34	20
SM3	1000	4360	4	0.495	4.36	20
SM4	1000	4316	3	0.466	4.31	22
SM5	1000	4520	7	0.492	4.52	20
BM1	1000	24564	1	0.592	24.56	5
BM2	1000	24636	1	0.427	24.36	4
BM3	1000	23476	1	0.274	23.47	4
BM4	1000	25066	1	0.246	25.06	4
BM5	1000	25106	1	0.148	25.10	4

3.1 Resolution Limit

We address the ring problem of Resolution Limit, a well-known problem in community detection. Quality-functions optimization is a key approach for community detection. Modularity aims to compute the inter-cluster number of edges inside a cluster and compare it with the expected number of such edges in a similar-sized random network. Modularity also computes the node degrees. This way, the randomness considers the possibility of any inter-node connection, which is impractical for large networks. Furthermore, the larger the network, the smaller the number of edges between two groups of nodes. So, if a network is large enough, the expected number of edges between two groups of nodes in modularity's null model may be smaller than one. Hence, modularity interprets a single edge as a sign of a strong correlation between the two clusters, and merging them would optimize modularity. In other words, for large networks, weak connections would merge for a dense graph [1]. The resolution problem is addressed by integrating nodes' influence on one another together with the flow propagation approach.

4 Experiment

4.1 Datasets and Setup

Two real-world benchmark datasets are used for this experiment: Polbooks: (a) A network of books about recent US politics sold by the on line bookseller Amazon.com. Edges between books represent frequent co-purchasing of books by the same buyers.

(b) Karate Club: social network of friendships between 34 members of a karate club at a US university in the 1970s [20].

We also used LFR benchmark datasets presented by Lancichinetti et al. [5]. LFR benchmark networks has features of real-world networks, the distributions of degree and the number of triangles in the network. Moreover, clustering coefficient and community sizes are adjustable with this benchmark network creator. The experimental environment includes Python 3.7 and NetworkX. The processor used to run the estimation and the proposed algorithm was Intel(R) Xeon(R) CPU E5-1630 v4 @ 3.70 GHz with a RAM of 64.00 GB on a Windows 10 pro 64-bit Operating System.

4.2 Results and Discussion

The goal of our computational experiment is to evaluate and validate the correctness and reliability of the BWLP algorithm. All the experiments are performed on both synthetic and real-world networks to ensure analyses of various distributions of nodes and edges. Detailed information about the datasets used is given in Sect. 4.1 as well as Tables 1 and 2.

Run time and accuracy are presented in Fig. 1. Figure 2 (a) and (b) shows the results for state-of-art-algorithms versus the proposed BWLP algorithm. Vertical line show the Normalized Mutual Information (NMI) [16] as an accuracy metric for benchmark graphs. The horizontal axes shows the mixing parameter μ . The two charts show community size ranges where (a) is from 10 to 50 vertices and (b) ranges from 20 to 100 vertices. Figure 2 (c) and (d) represent the results for state-of-art-algorithms versus the proposed BWLP algorithm. The vertical axes show the Adjusted Rand Index (ARI) [13]. The horizontal axes shows the mixing parameter μ .

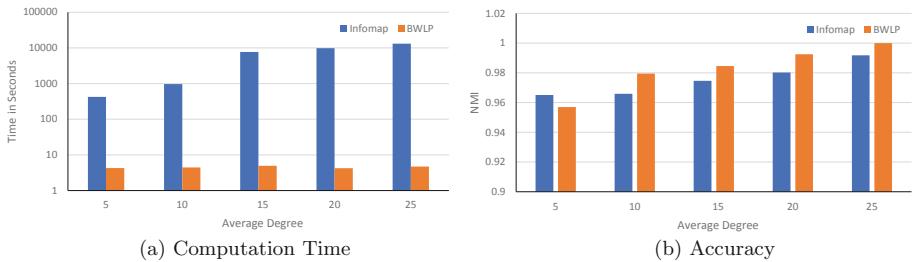
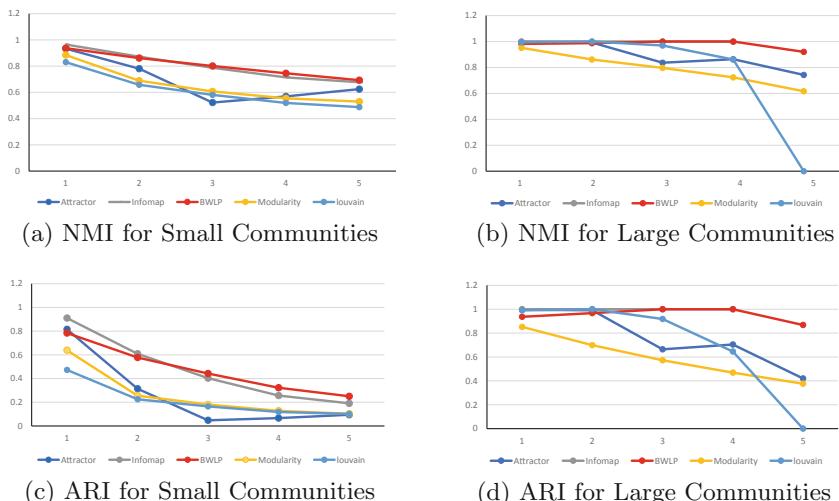
Figure 3.a and 3.b show the computational time of Louvain and Infomap against BWLP. Figure 3.a depicts BWLP achieving an average of 550 times faster run time than the benchmark algorithms. The ratio grows to 3500 times faster presented in Fig. 3.b. It should be noted that all the algorithms have been written in Python 3.7 and all experiments were run on the same machine.

As shown in Fig. 3, the computational times of Infomap and Louvain grow as the network gets larger. In these types of optimization methods, there are two problems: first is the hidden communities and resolution limit as explained in Sect. 3.1 which result in lower accuracy (shown in Fig. 1.b). The second is that the computational time grows polynomially as the number of edges in the dataset grows shown in Fig. 1.a).

Benchmark algorithms stated in Table 3 are not efficient as they calculate a quality score for every randomly chosen group of communities to measure their quality. Then, nodes will move between communities and with every move, the quality-function is re-calculated. If the quality score is better, the changes are kept; otherwise, changes are discarded. Infomap has a complexity of $O(n)$ in the best case, and $O(n^2 \log n)$ in the worst case. In addition to the complexity, Infomap has quality-function calculations with every adjustment, which makes the algorithm even slower. MCL has a complexity of $O(n^3)$ in all the cases which makes this algorithm slower than Infomap when it comes to big networks. In contrast, BWLP calculates a one-time weight value for all edges of every node

Table 3. Experiment results for given datasets

Datasets	Karate		polbooks	
	NMI	ARI	NMI	ARI
BWLP	1	1	0.568199	0.677826
Modularity	0.6924	0.6803	0.5308	0.6379
Infomap	0.699488	0.702155	0.493454	0.53606
Attractor	0.924092	0.939252	0.5308	0.6379
MCL	0.836498	0.882302	0.52086	0.59365

**Fig. 1.** (a) Run time (seconds) to detect the communities in LFR benchmark datasets based on varying node degrees. (b) Accuracy of the detected communities using NMI**Fig. 2.** NMI and ARI performances on the LFR benchmark.

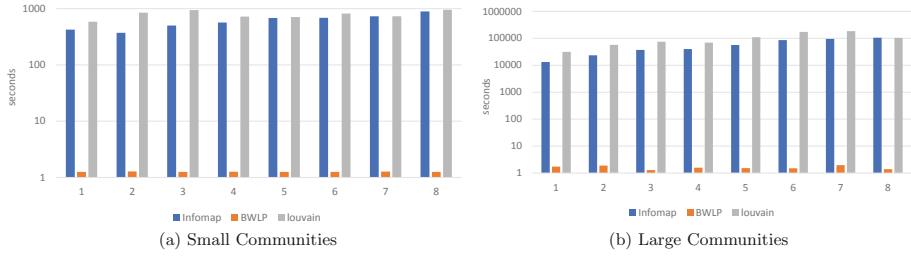


Fig. 3. Community detection run time (seconds) in LFR benchmark datasets with varying mixing parameter μ (horizontal axis)

and from there every node chooses its label (community group) based on its edge weight and neighboring labels. With this method, we avoid recalculating and regrouping of the nodes.

In terms of accuracy, Fig. 2.a, 2.b presents BWLP results over two networks of 1000 vertices with a mixing parameter μ in range of (1–5) on horizontal axis. Vertical axes shows the correctness measure of ARI [13]. BWLP outperforms all other algorithms except for the $\mu = 1$ in which Infomap is slightly better in Fig. 2.a $\mu = 1$. In the Fig. 2.b Louvain is slightly better only at $\mu = 1$.

In addition, Fig. 2.a and 2.b depict the correctness measure of Normalized Mutual Information (NMI) [16] on vertical axes and μ in range of (1–5) on the horizontal axis over two networks of 1000 vertices. The results from Fig. 2 confirm that the communities are detected with a higher accuracy by BWLP as compared to the other algorithms.

5 Conclusion

In this paper, a new multi-stage community detection algorithm is proposed. This algorithm automatically identifies the structures in networks based on dissimilarity between nodes and the structure of the network. There is no pre-processing or prior knowledge needed of the dataset/network for the proposed algorithm. We noticed that on average, nodes agree on the same label in less than three iterations.

The major phases of the proposed algorithm are threefold: The first is to identify the dissimilarity value and structural form of networks and to sort data nodes based on their dissimilarity values. The second layer is to find the flow of the network from each vertex, then re-label it to vertices which belong to that particular flow. Finally, the last layer converges all the nodes and construct the final community structures.

Extensive experiments were conducted to demonstrate the correctness and quality and expenses of the method introduced versus benchmark algorithms. Results shown in Sect. 4.2 demonstrate that BWLP requires lower resources as compared to benchmark algorithms. In addition, better accuracy is achieved particularly over the datasets with higher edge density and complexity. BWLP

successfully finds the hidden communities, a known problem for other benchmark algorithms. BWLP is a better choice for big data graphs considering it outperforms the state-of-the-art algorithms, being 550 to 3500 time faster for given data-sets as shown in Fig. 3(a) and Fig. 3(b). We presented the advantages of BWLP against several benchmark methods.

Future research plans are threefold: first, to extend the algorithm for weighted graphs; second, to find overlapping community structure with identical time and space complexity; and third, to improve the accuracy of the existing algorithms through introducing more conditions for special graph structures.

References

1. Fortunato, S., Barthélemy, M.: Resolution limit in community detection. *Proc. Natl. Acad. Sci.* **104**(1), 36–41 (2007). Resolution Limit
2. Girvan, M., Newman, M.E.J.: Community structure in social and biological networks. *Proc. Natl. Acad. Sci.* **99**(12), 7821–7826 (2002). Betweenness
3. Ioannidis, V.N., Zamzam, A.S., Giannakis, G.B., Sidiropoulos, N.D.: Coupled graph and tensor factorization for recommender systems and community detection. *IEEE Trans. Knowl. Data Eng.* (2019)
4. Lancichinetti, A., Fortunato, S., Kertész, J.: Detecting the overlapping and hierarchical community structure in complex networks. *New J. Phys.* **11**(3), 033015 (2009). LFM
5. Lancichinetti, A., Fortunato, S., Radicchi, F.: Benchmark graphs for testing community detection algorithms. *Phys. Rev. E* **78**, 046110 (2008)
6. Lee, C., Reid, F., McDaid, A., Hurley, N.: Detecting highly overlapping community structure by greedy clique expansion. arXiv preprint [arXiv:1002.1827](https://arxiv.org/abs/1002.1827) (2010). GCE
7. Mukunda, A., Pirouz, M.: Influence-based community detection and ranking. In: 2019 International Conference on Computational Science and Computational Intelligence (CSCI), pp. 1341–1346. IEEE (2019)
8. Palla, G., Derényi, I., Farkas, I., Vicsek, T.: Uncovering the overlapping community structure of complex networks in nature and society. arXiv preprint [physics/0506133](https://arxiv.org/abs/physics/0506133) (2005). CPM
9. Pirouz, M., Zhan, J.: Optimized label propagation community detection on big data networks. In: Proceedings of the 2018 International Conference on Big Data and Education, pp. 57–62 (2018)
10. Pirouz, M., Zhan, J., Tayeb, S.: An optimized approach for community detection and ranking. *J. Big Data* **3**(1), 1–12 (2016). <https://doi.org/10.1186/s40537-016-0058-z>
11. Psorakis, I., Roberts, S., Ebden, M., Sheldon, B.: Overlapping community detection using Bayesian non-negative matrix factorization. *Phys. Rev. E* **83**(6), 066114 (2011). NMF
12. Raghavan, U.N., Albert, R., Kumara, S.: Near linear time algorithm to detect community structures in large-scale networks. *Phys. Rev. E* **76**(3), 036106 (2007)
13. Rand, W.M.: Objective criteria for the evaluation of clustering methods. *J. Am. Stat. Assoc.* **66**(336), 846–850 (1971)
14. Shen, H., Cheng, X., Cai, K., Hu, M.-B.: Detect overlapping and hierarchical community structure in networks. *Physica A: Stat. Mech. Appl.* **388**(8), 1706–1712 (2009). EAGLE

15. Singh, S.S., Kumar, A., Singh, K., Biswas, B.: C2IM: community based context-aware influence maximization in social networks. *Physica A: Stat. Mech. Appl.* **514**, 796–818 (2019)
16. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **3**(Dec), 583–617 (2002). NMI
17. Li, W., Huang, C., Wang, M., Chen, X.: Stepping community detection algorithm based on label propagation and similarity. *Physica A: Stat. Mech. Appl.* **472**, 145–155 (2017)
18. Xie, J., Szymanski, B.K.: Community detection using a neighborhood strength driven label propagation algorithm. In: 2011 IEEE Network Science Workshop (NSW), pp. 188–195. IEEE (2011)
19. Xue, C., Wu, S., Zhang, Q., Shao, F.: An incremental group-specific framework based on community detection for cold start recommendation. *IEEE Access* **7**, 112363–112374 (2019)
20. Zachary, W.W.: An information flow model for conflict and fission in small groups. *J. Anthropol. Res.* **33**(4), 452–473 (1977). <http://www-personal.umich.edu/~mejn/netdata/>
21. Zhang, P., Moore, C., Newman, M.E.J.: Community detection in networks with unequal groups. *Phys. Rev. E* **93**(1), 012303 (2016)
22. Zhang, S., Wang, R.-S., Zhang, X.-S.: Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A: Stat. Mech. Appl.* **374**(1), 483–490 (2007). Spectral, Euclidean Space, Fuzzy C-mean Algorithm
23. Zhou, D., Wang, X.: A neighborhood-impact based community detection algorithm via discrete PSO. *Math. Prob. Eng.* **2016** (2016)



Analysis of the Vegetation Index Dynamics on the Base of Fuzzy Time Series Model

Elchin Aliyev, Ramin Rzayev^(✉), and Fuad Salmanov

Institute of Control Systems of ANAS, B. Vahabzadeh str. 9, 1141 Baku, Azerbaijan

Abstract. Based on the fuzzy analysis of satellite monitoring data, the annual dynamics of the vegetation index for the selected crop area is investigated by means of MODIS images (LPDAAC – the Land Processes Distributed Active Archive Center). To reconstruct and predict the weakly structured vegetation index time series, the fuzzy models are proposed, compiled taking into account the analysis of internal connections of the first and second orders, which are presented in the form of fuzzy relations. The proposed models were investigated for adequacy and suitability from the point of view of the analysis of the peculiarities of the intra-annual mean annual dynamics of the index, typical for the cultivated area. On the basis of the proposed approach, the results of the study of long-term dynamics of vegetation indices can be used for a complex analysis of the dynamics of vegetation cover, including modeling and forecasting the efficiency and productivity of agricultural crops.

Keywords: Vegetation index · Fuzzy time series · Fuzzy relation

1 Introduction

Modern technologies for satellite monitoring of the Earth's surface provide agricultural producers with useful information about the health of crops. The remote sensor's ability to detect minor differences in vegetation makes it a useful tool for quantifying variability within a given field, assessing crop growth, and managing land based on current conditions. Remote sensing data, collected on a regular basis, allows growers and agronomists to make a current map of the state and strength of crops, analyze the dynamics of changes in the health of crops, and predict it. For the interpretation of remote sensing data, the most effective means are all kinds of vegetation indices, in particular, Normalized Difference Vegetation Index (NDVI), which are calculated empirically, i.e. by operations with different spectral ranges of satellite monitoring data.

Most agricultural crops are characterized by changes in the phases of development, which is reflected in the dynamics of the spectral-reflective properties of plants. The study of seasonal and long-term changes in the spectral and brightness characteristics of crops is possible through the analysis and modeling of the dynamic series of vegetation indices, which makes it possible to quantitatively evaluate the features of the vegetation cover and the patterns of its temporal dynamics. At the same time, standard algorithms for solving problems of predicting the dynamics of the spectral-reflective properties

of plants work, as a rule, with “crisp” or structured data of satellite earth sensing, i.e. with data presented as averaged numbers. Therefore, the averaging of the results of measurements of vegetation indices is one of the most common empirical operations in systems for collecting data from satellite monitoring and crop management. In particular, the achievement of the required accuracy in the NDVI averaging process is achieved by multiple measurements, where the results of individual measurements are partially compensated by positive and negative deviations from the exact value. The accuracy of their mutual compensation improves with an increase in the number of measurements, since the absolute value of the mean of negative deviations approaches to the mean of positive deviations.

More generally, satellite monitoring data, for example, NDVI values should be considered as weakly structured, i.e. those about which only their belonging to a certain type is known [1]. In particular, the interval $[NDVI_{min}, NDVI_{max}]$, which includes all its measurements, can serve as an adequate reflection of the poorly structured NDVI values. Another more adequate reflection of the weakly structured value of NDVI can be a statement of the form “HIGH”, which, in fact, is one of the terms (values) of the linguistic variable “*value of the vegetation index*”, which can be formally described by the appropriate fuzzy set [2]. Therefore, based on this premise, it becomes obvious the importance and relevance of studying methods for studying seasonal and long-term changes in the spectral-brightness characteristics of crops by means of time series of satellite monitoring indicators relative to NDVI, which we will consider as weakly structured values.

2 Problem Definition

On the basis of reliable information obtained by remote sensing of the Earth, it is necessary to assess the regularities in the change in the average long-term values of NDVI on the specifically selected area of cropland. More generally, the study of the NDVI dynamics should include three stages: 1) calculation and analysis of the annual dynamics of the NDVI values averaged for each analyzed date, typical for the selected region; 2) analysis of the time series of long-term NDVI values; 3) estimation of the dynamics of the average annual NDVI values for cultivated areas for a certain period. The purpose of this study is to provide a fuzzy analysis of seasonal and long-term characteristics in the NDVI dynamics for the cultivated areas created by the MODIS platform (LPDAAC) [3].

3 Fuzzy Modeling of NDVI Dynamics

The existing methods of fuzzy modeling of weakly structured time series imply the sequential implementation of the following steps: 1) building the coverage of the entire data set in the form of a universal set (universe); 2) fuzzification of weakly structured time series data; 3) definition of internal connections in the form of fuzzy relations and their division into groups; 4) determination the fuzzy outputs of the applied model and their defuzzification.

To define a universe, the following step-by-step procedure is applied [4].

Step 1. Sorting the time series data $\{x_t\}$ ($t = 1 \div n$) into an ascending sequence $\{x_{p(i)}\}$, where p is a permutation that sorts the data values in ascending order, i.e. $x_{p(t)} \leq x_{p(t+1)}$.
 Step 2. Calculation of the mean value over the set of all pairwise distances $d_i = |x_{p(i)} - x_{p(i+1)}|$ between any two consecutive values $x_{p(i)}$ and $x_{p(i+1)}$ by the formula:

$$AD(d_1, d_2, \dots, d_n) = \frac{1}{n-1} \sum_{i=1}^{n-1} |x_{p(i)} - x_{p(i+1)}|, \quad (1)$$

and the standard deviation by the formula

$$\sigma_{AD} = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n-1} (d_i - AD)^2}. \quad (2)$$

Step 3. Detection and elimination of anomalies – sharply distinguished quantities to be ejected. It uses both the mean distance AD and the standard deviation σ_{AD} from the previous step. In this case, the values of pairwise distances that do not satisfy the condition are subject to ejection:

$$AD - \sigma_{AD} \leq d_i \leq AD + \sigma_{AD}. \quad (3)$$

Step 4. Recalculate the mean distance between any two consecutive values from the set of values remaining after sorting for outliers.

Step 5. Establishing the universe U in the form of $U = [D_{\min} - AD, D_{\max} + AD] = [D_1, D_2]$, where D_{\min} and D_{\max} are the minimum and maximum values, respectively, on the entire data set $\{x_t\}$ ($t = 1 \div n$).

There are various ways to identify membership functions that restore the fuzzy subsets of the given universe. In particular, one of such methods is symmetric trapezoidal membership functions of the following form (see Fig. 1):

$$\mu_{A_k}(x) = \begin{cases} 0, & x < a_{k1} \\ \frac{x-a_{k1}}{a_{k2}-a_{k1}}, & a_{k1} \leq x \leq a_{k2}, \\ 1, & a_{k2} \leq x \leq a_{k3}, \\ \frac{a_{k4}-x}{a_{k4}-a_{k3}}, & a_{k3} \leq x \leq a_{k4}, \\ 0, & x > a_{k4}, \end{cases} \quad (4)$$

with parameters satisfying the conditions: $a_{k2} - a_{k1} = a_{k3} - a_{k2} = a_{k4} - a_{k3}$, where $k = 1 \div m$, m is the total number of fuzzy sets A_k describing the time series data. According to [4], this number is calculated by the formula:

$$m = [D_2 - D_1 - AD]/[2 \cdot AD]. \quad (5)$$

As an example, it was chosen the time series reflecting the annual dynamics of the NDVI index (see Table 1) based on MODIS (LPDAAC) images (Fig. 2, [5]) of the cultivated area in Jonesboro (USA) with geographic coordinates $(-90.1614583252562, 35.8135416634583)$.

For the entire set of time series data, after recalculating the average distance and standard deviation according to formulas (1) and (2) at the 5th step their final values are

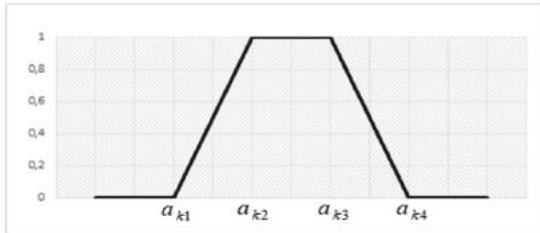


Fig. 1. Symmetric trapezoidal membership function.

Table 1. NDVI time series.

Nº	Date	NDVI	Nº	Date	NDVI
1	18.02.2000	0.3599	9	25.06.2000	0.7101
2	05.03.2000	0.4099	10	11.07.2000	0.7135
3	21.03.2000	0.368	11	27.07.2000	0.2479
4	06.04.2000	0.3296	12	12.08.2000	0.6587
5	22.04.2000	0.2535	13	28.08.2000	0.5473
6	08.05.2000	0.2966	14	13.09.2000	0.4815
7	24.05.2000	0.3211	15	29.09.2000	0.3719
8	09.06.2000	0.5104	16	15.10.2000	0.3217

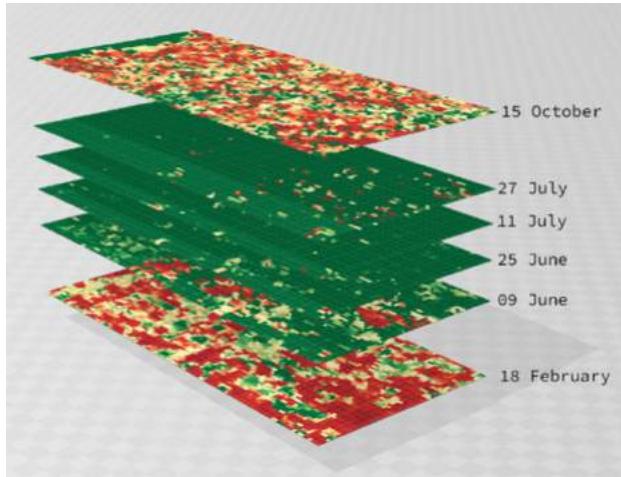


Fig. 2. MODIS (LPDAAC) images.

established under fulfilling condition (3): $AD = 0.0296$ and $\sigma_{AD} = 0.0007$. In this case, the desired universe is defined as the following segment $U = [0.2479 - 0.0296, 0.7135 +$

$0.0296] = [0.2183, 0.7431]$, where 0.2479 and 0.7135 are the minimum and maximum values of the NDVI, respectively. To describe the qualitative criteria for evaluating the NDVI the sufficient number of fuzzy subsets of the U is established from equality (4) as follows: $m = [0.7431 - 0.2183 - 0.0296]/[2 \cdot 0.0296] = 8.3649 \approx 8$. Then the corresponding 8 trapezoidal membership functions are identified (see Fig. 3), the parameters of which are summarized in Table 2.

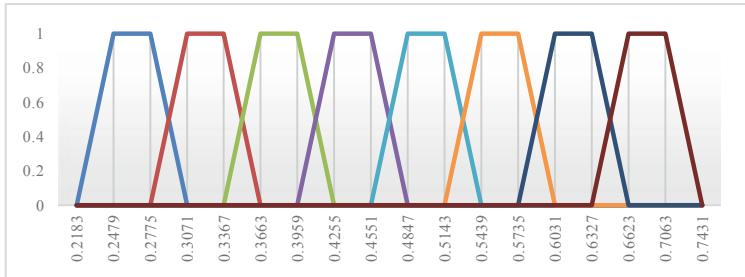


Fig. 3. Trapezoidal membership functions describing time series data.

Table 2. Parameters of trapezoidal membership functions.

Fuzzy set	Parameters			
	a_{k1}	a_{k2}	a_{k3}	a_{k4}
A_1	0.2183	0.2479	0.2775	0.3071
A_2	0.2775	0.3071	0.3367	0.3663
A_3	0.3367	0.3663	0.3959	0.4255
A_4	0.3959	0.4255	0.4551	0.4847
A_5	0.4551	0.4847	0.5143	0.5439
A_6	0.5143	0.5439	0.5735	0.6031
A_7	0.5735	0.6031	0.6327	0.6623
A_8	0.6327	0.6695	0.7063	0.7431

According to [4], fuzzification of time series data by identified trapezoidal membership functions is carried out by following rule: NDVI is described by those fuzzy set to which its value belongs with the greatest degree. When the NDVI value belongs to the interval $[a_{k2}, a_{k3}]$, it is relatively easy to find its fuzzy analogue. In other cases, clarifications are needed. In particular, according to (4) for $\text{NDVI} = 0.3599$ we have: $\mu_{A_3}(0.3599) = 0.7838$ and $\mu_{A_2}(0.3599) = 0.2162$ (see Fig. 4). Therefore, fuzzy set A_3 should be chosen as an analogue, because the value of its membership function at the point 0.3599 is greater. The obtained fuzzy analogs for all NDVI indexes are summarized in Table 3.

Fuzzy time series modeling is created on the base of the analysis of internal relationships (causal-effect relations) of different orders between the NDVI values. Within the

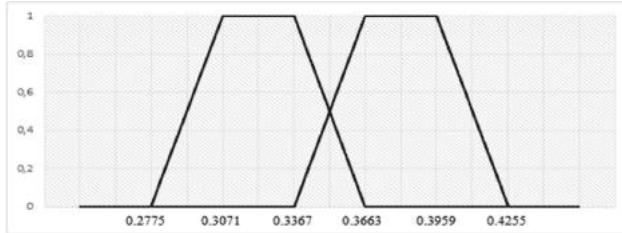


Fig. 4. Neighboring membership functions.

Table 3. Fuzzy time series.

Nº	Date	NDVI	Fuzzy set	Nº	Date	NDVI	Fuzzy set
1	18.02.2000	0.3599	A ₃	9	25.06.2000	0.7101	A ₈
2	05.03.2000	0.4099	A ₄	10	11.07.2000	0.7135	A ₈
3	21.03.2000	0.3680	A ₃	11	27.07.2000	0.2479	A ₁
4	06.04.2000	0.3296	A ₂	12	12.08.2000	0.6587	A ₈
5	22.04.2000	0.2535	A ₁	13	28.08.2000	0.5473	A ₆
6	08.05.2000	0.2966	A ₂	14	13.09.2000	0.4815	A ₅
7	24.05.2000	0.3211	A ₂	15	29.09.2000	0.3719	A ₃
8	09.06.2000	0.5104	A ₅	16	15.10.2000	0.3217	A ₂

framework of the NDVI fuzzy time series, the internal relationships of the 1st and 2nd orders were chosen. These relationships demonstrate fuzzy relations in the implicative form of the form “If <...>, then <...>” [6]. For example, internal relationships of the 1st order are grouped according to the principle: if the fuzzy set A₃ is related to the sets A₄ and A₂, then the group of the 1st order is localized relative to it: A₃ ⇒ A₂, A₄ (see Table 4, group G3). The groups of the 1st and 2nd orders are summarized in Table 4 and Table 5, respectively.

Table 4. The groups of the 1st order relationships

Group	Relation	Group	Relation	Group	Relation	Group	Relation
G1	A ₁ ⇒ A ₂ , A ₈	G3	A ₃ ⇒ A ₂ , A ₄	G5	A ₅ ⇒ A ₃ , A ₈	G7	A ₈ ⇒ A ₁ , A ₆ , A ₈
G2	A ₂ ⇒ A ₁ , A ₂ , A ₅	G4	A ₄ ⇒ A ₃	G6	A ₆ ⇒ A ₅		

If the NDVI value on the *i*-th day is denoted by x_i , and the NDVI value on the next (*i* + 1)-th day is denoted by x_{i+1} , then the 1st order internal relationship, for example, A₄

Table 5. The groups of the 2nd order relationships.

Group	Relation	Group	Relation	Group	Relation
G1	$A_3, A_4 \Rightarrow A_3$	G6	$A_2, A_2 \Rightarrow A_5$	G11	$A_1, A_8 \Rightarrow A_6$
G2	$A_4, A_3 \Rightarrow A_2$	G7	$A_2, A_5 \Rightarrow A_8$	G12	$A_8, A_6 \Rightarrow A_5$
G3	$A_3, A_2 \Rightarrow A_1$	G8	$A_5, A_8 \Rightarrow A_8$	G13	$A_6, A_5 \Rightarrow A_3$
G4	$A_2, A_1 \Rightarrow A_2$	G9	$A_8, A_8 \Rightarrow A_1$	G14	$A_5, A_3 \Rightarrow A_2$
G5	$A_1, A_2 \Rightarrow A_2$	G10	$A_8, A_1 \Rightarrow A_8$		

$\Rightarrow A_3$ can be interpreted as a fuzzy implicative rule: “If $x_i = A_4$, then $x_{i+1} = A_3$ ”. Or, for example, the 1st order internal relationship of the form $A_8 \Rightarrow A_1, A_6, A_8$ can be interpreted as a fuzzy implicative rule: “If $x_i = A_8$, then $x_{i+1} = A_1$ OR $x_{i+1} = A_6$ OR $x_{i+1} = A_8$ ”. Accordingly, the 2nd order internal relationship, for example, $A_3, A_4 \Rightarrow A_3$ can be interpreted as: “If $x_i = A_3$ AND $x_i = A_4$, then $x_{i+1} = A_3$ ”.

Various models are used to define fuzzy forecasts and their defuzzification. As one of these, a model was chosen, the essence of which is as follows [7]. If NDVI x_i is described by the fuzzy set A_j , which within the totality of all data forms only one internal relationship, for example, the relation: $A_j \Rightarrow A_k$, then the forecast for the next $(i + 1)$ -th day is the fuzzy set A_k . In the case, when there is the group of internal relationships, for example, $A_j \Rightarrow A_{k1}, A_{k2}, \dots, A_{kp}$, then the union $A_{k1} \cup A_{k2} \cup \dots \cup A_{kp}$ is the fuzzy forecast for the $(i + 1)$ th day. To defuzzify the outputs of this model, the following two principles are applied [1, 6, 7].

Principle 1 [1, 6]. In the case of the fuzzy relation of the form $A_i \Rightarrow A_j$, where A_i is the fuzzy analogue of NDVI on the i -th day, the crisp forecast for the next $(i + 1)$ -th day as the defuzzified value of the fuzzy forecast A_j , is the abscissa of the bisecting point of the upper base of the corresponding trapezoid. Actually, according to

$$F(A) = \frac{1}{\alpha_{\max}} \int_0^{\alpha_{\max}} M(A_\alpha) d\alpha, \quad (6)$$

where $A_\alpha = \{u | \mu_A(u) \geq \alpha, u \in U\}$ are the α -level sets ($\alpha \in [0, 1]$); $M(A_\alpha) = \sum_{k=1}^m u_k/m$ ($u_k \in A_\alpha$) are the cardinalities of the corresponding α -level sets, for the fuzzy set $A_3 = \{0/0.3367, 1/0.3663, 1/0.3959, 0/0.4255\}$ (see Table 2), which is a forecast in the fuzzy relation $A_4 \Rightarrow A_3$, we have:

For $0 < \alpha < 1$, $\Delta\alpha = 1$, $A_{3\alpha} = \{0.3663, 0.3959\}$, $M(A_{3\alpha}) = (0.3663 + 0.3959)/2 \approx 0.3811$.

Then, according to (5), the crisp (defuzzified) output of the model is calculated as

$$F(A_3) = \frac{1}{1} \int_0^1 M(A_{3\alpha}) d\alpha \approx M(A_{3\alpha}) \cdot \Delta\alpha = 0.3811.$$

Principle 2 [7]. In the case of the fuzzy relation of the form $A_i \Rightarrow A_j, A_t, A_p$, where A_i is the fuzzy analogue of NDVI on the i -th day, the crisp forecast for the next $(i + 1)$ -th day

is calculated as the arithmetic mean of the abscissa of the midpoints of the upper bases of trapeziums corresponding to the fuzzy sets A_j , A_t and A_p . In particular, according to the group of internal relationships $A_8 \Rightarrow A_1, A_6, A_8$ the forecast F for 11.07.2000, 27.07.2000 and 28.08.2000 dates is calculated as follows:

$$F = [(0.2479 + 0.2775)/2 + (0.5439 + 0.5735)/2 + (0.6695 + 0.7063)/2]/3 = 0.5031.$$

The forecasts obtained on the basis of the 1st order predictive model are summarized in Table 6, and the geometric interpretation of this model is shown in Fig. 5.

Table 6. The 1st order predictive model.

Date	NDVI	Output	Predict	Date	NDVI	Output	Predict
18.02.2000	0.3599			18.02.2000	0.3599	A_3, A_8	0.5345
05.03.2000	0.4099	A_2, A_4	0.3811	05.03.2000	0.4099	A_1, A_6, A_8	0.5031
21.03.2000	0.3680	A_3	0.3811	21.03.2000	0.3680	A_1, A_6, A_8	0.5031
06.04.2000	0.3296	A_2, A_4	0.3811	06.04.2000	0.3296	A_2, A_8	0.5049
22.04.2000	0.2535	A_1, A_2, A_5	0.3614	22.04.2000	0.2535	A_1, A_6, A_8	0.5031
08.05.2000	0.2966	A_2, A_8	0.5049	08.05.2000	0.2966	A_5	0.4995
24.05.2000	0.3211	A_1, A_2, A_5	0.3614	24.05.2000	0.3211	A_3, A_8	0.5345
09.06.2000	0.5104	A_1, A_2, A_5	0.3614	09.06.2000	0.5104	A_2, A_4	0.3811

The set of internal connections of the 2nd and higher orders coincides with the set of NDVI fuzzy analogs (see Table 3). Nevertheless, we considered it necessary to show the defuzzified outputs of the 2nd order fuzzy model (see Table 7) and its geometric interpretation (see Fig. 5).

Table 7. The 2nd order predictive model.

Date	NDVI	Output	Predict	Date	NDVI	Output	Predict
18.02.2000	0.3599			18.02.2000	0.3599	A_8	0.6879
05.03.2000	0.4099			05.03.2000	0.4099	A_8	0.6879
21.03.2000	0.3680	A_3	0.3811	21.03.2000	0.3680	A_1	0.2627
06.04.2000	0.3296	A_2	0.3219	06.04.2000	0.3296	A_8	0.6879
22.04.2000	0.2535	A_1	0.2627	22.04.2000	0.2535	A_6	0.5587
08.05.2000	0.2966	A_2	0.3219	08.05.2000	0.2966	A_5	0.4995
24.05.2000	0.3211	A_2	0.3219	24.05.2000	0.3211	A_3	0.3811
09.06.2000	0.5104	A_5	0.4995	09.06.2000	0.5104	A_2	0.3219

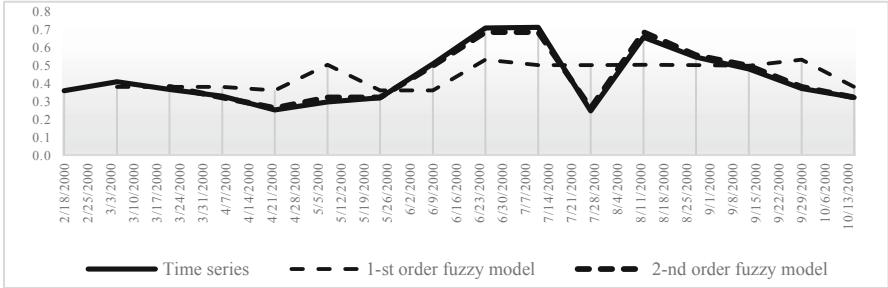


Fig. 5. Fuzzy time series models.

To assess the adequacy of fuzzy time series models the following criteria are used:

$$\text{MAPE} = \frac{1}{m} \sum_{j=1}^m \frac{|F_j - A_j|}{A_j} \times 100, \quad \text{MSE} = \frac{1}{m} \sum_{j=1}^m (F_j - A_j)^2,$$

where A_j and F_j denote the actual value and predict at the t_j date, respectively. The results of forecasting and assessments of their reliability for both models (see Fig. 5) are summarized in Table 8.

Table 8. Assessments of the model reliabilities.

Date	NDVI	Fuzzy models		Date	NDVI	Fuzzy models	
		1 st order	2 nd order			1 st order	2 nd order
18.02.2000	0.3599			18.02.2000	0.3599	0.5345	0.6879
05.03.2000	0.4099	0.3811		05.03.2000	0.4099	0.5031	0.6879
21.03.2000	0.3680	0.3811	0.3811	21.03.2000	0.3680	0.5031	0.2627
06.04.2000	0.3296	0.3811	0.3219	06.04.2000	0.3296	0.5049	0.6879
22.04.2000	0.2535	0.3614	0.2627	22.04.2000	0.2535	0.5031	0.5587
08.05.2000	0.2966	0.5049	0.3219	08.05.2000	0.2966	0.4995	0.4995
24.05.2000	0.3211	0.3614	0.3219	24.05.2000	0.3211	0.5345	0.3811
09.06.2000	0.5104	0.3614	0.4995	09.06.2000	0.5104	0.3811	0.3219
MSE						0.0186	0.0198
MAPE						30.5881	3.2796

4 Conclusion

The results of the study of NDVI long-term dynamics can be used for modeling and forecasting the effectivity and productivity of agricultural crops, for the complex analysis

of the vegetation dynamics. Certainly, the obtained fuzzy models cannot be a reliable source of information for conclusions about the prospects for the state of vegetation on a given plot of agricultural land, because, firstly, they rely on a small amount of annual NDVI data, which is extremely insufficient for analyzing internal cause-effect relations of higher orders, and, secondly, they describe one annual cycle. Nevertheless, this approach can be extrapolated to all long-term NDVI data obtained using MODIS technology. As a result, it is possible to model and accordingly predict, for example, the dynamics of long-term seasonal NDVI values averaged for each analyzed date, as well as the dynamics of the average annual, annual minimum and annual maximum NDVI values.

References

1. Rzayev, R.R.: Analytical Support for Decision-Making in Organizational Systems. Palmerium Academic Publishing, Saarbruchen (2016). (in Russian)
2. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning. *Inf. Sci.* **8**(3), 199–249 (1965)
3. Vegetation Indices 16-Day L3 Global 250 m MOD13Q1. LPDAAC. https://lpdaac.usgs.gov/dataset_discovery/modis/modis_products_table/mod13q1. Accessed 12 Oct 2020
4. Ortiz-Arroyo, D., Poulsen, J.R.: A weighted fuzzy time series forecasting model. *Indian J. Sci. Technol.* **11**(27), 1–11 (2018)
5. Vegetation Indices 16-Day L3 Global 250 m MOD13Q1. LPDAAC: <https://goo.gl/maps/YAdomuoXsD4QQN36>. Accessed 12 Oct 2020
6. Rzayev, R.R., et al.: Time series modeling based on fuzzy analysis of position-binary components of historical data. *Nechetkie Sistemy i Myagkie Vychisleniya* [Fuzzy Systems and Soft Computing] **10**(1), 35–73 (2015). (in Russian)
7. Chen, S.M.: Forecasting enrollments based on high-order fuzzy time series. *Cybern. Syst. Int. J.* **33**, 1–16 (2002)



Application of Data–Driven Fault Diagnosis Design Techniques to a Wind Turbine Test–Rig

Silvio Simani¹(✉), Saverio Farsoni¹, and Paolo Castaldi²

¹ Department of Engineering, University of Ferrara, Ferrara, Italy
silvio.simani@unife.it

² Department of Electrical, Electronic, and Information Engineering,
University of Bologna, Bologna, Italy
<http://www.silviosimani.it>

Abstract. The fault diagnosis of safety critical systems such as wind turbine installations includes extremely challenging aspects that motivate the research issues considered in this paper. In particular, this work studies fault diagnosis solutions that are considered in a viable way and used as advanced techniques for condition monitoring of dynamic processes. To this end, the work proposes the design of a fault diagnosis strategies that exploits the estimation of the fault by means of data–driven approaches. This solution leads to the development of effective methods allowing the management of partial unknown information of the system dynamics, while coping with measurement errors, the model–reality mismatch and other disturbance effects. In mode detail, the proposed data–driven methodologies exploit fuzzy systems and neural networks in order to estimate the nonlinear dynamic relations between the input and output measurements of the considered process and the faults. To this end, the fuzzy and neural network structures are integrated with auto–regressive with exogenous input descriptions, thus making them able to approximate unknown nonlinear dynamic functions with arbitrary degree of accuracy. Once these models are estimated from the input and output data measurement acquired from the considered dynamic process, the capabilities of their fault diagnosis capabilities are validated by using a high–fidelity benchmark that simulates the healthy and the faulty behaviour of a wind turbine system. Moreover, at this stage the benchmark is also useful to analyse the robustness and the reliability characteristics of the developed tools in the presence of model–reality mismatch and modelling error effects featured by the wind turbine simulator. On the other hand, a hardware–in–the–loop tool is finally implemented for testing the performance of the developed fault diagnosis strategies in a more realistic environment.

Keywords: Wind turbine · Data–driven approach · Fuzzy Systems · Neural Networks · Hardware–in–the–loop

1 Introduction

As the power required worldwide is increasing, and at the same time in order to meet low carbon requirements, one of the possible solutions consists of increasing the exploitation of wind-generated energy. However, this need implies to improve the levels of availability and reliability, which represent the fundamental ‘sustainability’ feature, extremely good for these renewable energy conversion systems. In fact, wind turbine processes should produce the required amount of electrical energy continuously and in an effective way, trying to maximise the wind power capture, on the basis of the grid’s demand and despite malfunctions. To this end, possible faults affecting this energy conversion system must be properly diagnosed and managed, in their earlier occurrence, before they degrade into failures and become critical issues.

Another important aspect regards wind turbines with very large rotors, which allow to reach megawatt size, possibly maintaining light load carrying structures. On one hand, they become very expensive and complex plants, which require advanced control techniques to reduce the effects of induced torques and mechanical stress on the structures. On the other hand, they need an extremely high level of availability and reliability, in order to maximise the generated energy, to minimise the production cost, and to reduce the requirements of Operation and Maintenance (O & M) services. In fact, the final cost of the produced energy depends first on the installation expenses of the plant (fixed cost). Second, unplanned O&M costs may increase it up to about 30%, especially if offshore wind turbine installations are considered, as they work in harsh environments.

These considerations motivate the implementation of condition monitoring and fault diagnosis techniques that can be integrated into fault tolerant control strategies, *i.e.* the ‘sustainable’ solutions. Unfortunately, several wind turbine manufacturers do not implement ‘active’ approaches against faults, but rather adopt conservative solutions. For example, this means to resort to the shutdown of the plant and wait for O&M service. Therefore, more effective tools for managing faults in an active way must be considered, in order to improve the wind turbine working conditions, and not only during faulty behaviour.

Note also that, in principle, this strategy will be able to prevent critical faults that might affect other components of the wind turbine system, and may thus avoid to require unplanned replacement of its functional parts. Moreover, it will lead to decrease possible O&M costs, while improving the energy production and reducing the energy cost. On the other hand, the implementation of advanced control systems, with the synergism with big data tools and artificial intelligence techniques will lead to the development of real-time condition monitoring, fault diagnosis and fault tolerant control solutions for these safe-critical systems that can be enabled also only with on-demand features.

In the last decades several papers have investigated the problem of fault diagnosis for wind turbine systems, as addressed *e.g.* in [6]. Some of them have investigated the diagnosis of particular faults, *i.e.* those affecting the drive-train part of the wind turbine nacelle. In fact, sometimes the detection of these faults can be enhanced if the wind turbine subsystems are compared to other

modules of the whole plant [9]. Moreover, more challenging topics regarding the fault tolerant control of wind turbines have been considered *e.g.* in [11], also by proposing international co–operations on common problems, as analysed *e.g.* in [7]. Therefore, the key point is represented by the fault diagnosis task when exploited to achieve the sustainability feature for safety–critical systems, such as wind turbines. In fact, it has been shown to represent a challenging topic [20], thus justifying the research issues investigated in this paper.

With reference to this paper, the topic of the fault diagnosis of a wind turbine system is analysed. In particular, the design of practical and reliable solutions to Fault Detection and Isolation (FDI) is considered. However, differently from other works by the same authors, the Fault Tolerant Control (FTC) topic is not investigated here, even if it can rely on the same tools exploited in this paper. In fact, the fault diagnosis module provides the reconstruction of the fault signal affecting the process, which could be actively compensated by means of a controller accommodation mechanism. Moreover, the fault diagnosis design is enhanced by the derived fault reconstructors that are estimated via data–driven approaches, as they also allow to accomplish the fault isolation task.

The first data–driven strategy proposed in this work exploits Takagi–Sugeno (TS) fuzzy prototypes [1,4], which are estimated via a clustering algorithm and exploiting the data–driven algorithm developed in [15]. For comparison purpose, a further approach is designed, which exploits Neural Networks (NNs) to derive the nonlinear dynamic relations between the input and output measurements acquired for the process under diagnosis and the faults affecting the plant. The selected structures belong to the feed–forward Multi–Layer Perceptron (MLP) neural network class that include also Auto–Regressive with eXogenous (ARX) inputs in order to model nonlinear dynamic links among the data. In this way, the training of these Nonlinear ARX (NARX) prototypes for fault estimation can exploit standard back–propagation training algorithm, as recalled *e.g.* in [5].

The designed fault diagnosis schemes are tested via a high–fidelity simulator of a wind turbine process, which describes its behaviour in healthy and faulty conditions. This simulator, which represents a benchmark [8], includes the presence of uncertainty and disturbance effects, thus allowing to verify the reliability and robustness characteristics of the proposed fault diagnosis methodologies. Moreover, this work proposes to validate the efficacy of the designed fault diagnosis techniques by exploiting a more realistic scenario, which consists of a Hardware–In–the–Loop (HIL) tool.

It is worth noting the main contributions of this paper with respect to previous works by the authors. For example, this study analyses the solutions addressed *e.g.* in [19] but taking into account a more realistic and real–time system illustrated in Sect. 4. On the other hand, the fault diagnosis scheme developed in this paper was designed for a wind turbine system also in [17], but without considering the HIL environment.

The fuzzy methodology was also proposed in by the authors in [14], which considered the development of recursive algorithms for the implementation of adaptive laws relying on Linear Parameter Varying (LPV) systems. The app-

roach proposed in this paper estimates the fault diagnosis models by means of off-line procedures. Moreover, this paper further develops the achievements obtained *e.g.* in [18], but concerning the fault diagnosis a wind farm. The paper [16] proposed the design of a fault tolerant controller using the input-output data achieved from a single wind turbine, by exploiting the results achieved in [17]. On the other hand, this work considers the verification and the validation of the proposed fault diagnosis methodologies by exploiting an original HIL tool, proposed considered in a preliminary paper by the same authors [13].

The work follows the structure sketched in the following. Section 2 briefly summarises the wind turbine simulator, as it represents a well-established benchmark available in literature [10]. Section 3 describes the fault diagnosis strategies based on Fuzzy Systems (FSs) and NN structures. Section 4 summarises the obtained results via the simulations and the HIL tool describing the behaviour of the wind turbine process. Finally, Sect. 5 concludes the work by reporting the main points of the paper and suggesting some interesting issues for further research and future investigations.

2 Wind Turbine System Description

The Wind Turbine (WT) benchmark considered in this work for validation purposes was earlier presented in [8, 10] and motivated by an international competition. Despite its quite simple structure, it is able to describe quite accurately the actual behaviour of a three-blade horizontal-axis wind turbine that is working at variable-speed and it is controlled by means of the pitch angle of its blades. The plant includes several interconnected subsystems, namely the wind process, the wind turbine aerodynamics, the drive-train, the electric generator/converter, the sensor and actuator systems and the baseline controller. The overall system is sketched in Fig. 1, which represents the fault diagnosis target developed in this work. Further details of the WT benchmark will not be provided here, as they were described in detail in [9] and the references therein.

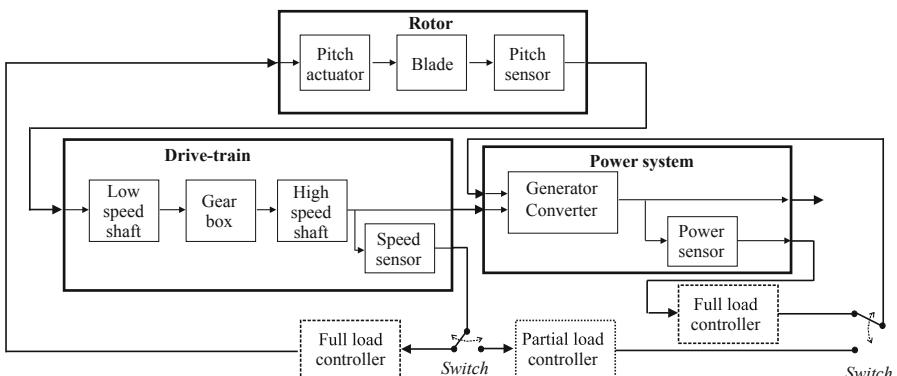


Fig. 1. The WT benchmark and its functional subsystems.

This wind turbine benchmark is able to generate different typical fault cases affecting the sensors, the actuators and the process components. This scenario comprising 9 fault situations is illustrated by means of Table 1, which reports the input and output measurements acquired from the WT process signals and mainly affected by these faults.

Table 1. Fault scenario of the WT benchmark.

Fault case	Fault type	Most affected input–output measurements
1	Sensor	$\beta_{1,m1}, \beta_{1,m2}, \omega_{g,m2}$
2	Sensor	$\beta_{1,m2}, \beta_{2,m2}, \omega_{g,m2}$
3	Sensor	$\beta_{1,m2}, \beta_{3,m1}, \omega_{g,m2}$
4	Sensor	$\beta_{1,m2}, \omega_{g,m2}, \omega_{r,m1}$
5	Sensor	$\beta_{1,m2}, \omega_{g,m2}, \omega_{r,m2}$
6	Actuator	$\beta_{1,m2}, \beta_{2,m1}, \omega_{g,m2}$
7	Actuator	$\beta_{1,m2}, \beta_{3,m2}, \omega_{g,m2}$
8	Actuator	$\beta_{1,m2}, \tau_{g,m}, \omega_{g,m2}$
9	System	$\beta_{1,m2}, \omega_{g,m1}, \omega_{g,m2}$

In this way, Table 1 reports the most sensitive measurements $u_j(k)$ and $y_l(k)$ acquired from the WT system with respect to the fault conditions implemented in the WT benchmark. In practice, the fault signals of Table 1 were injected into the WT simulator, assuming that only a single fault may occur. Then, by checking the Relative Mean Square Errors (RMSEs) between all the fault-free and faulty measurements from the WT plant, the most sensitive signal $u_j(k)$ and $y_l(k)$ was selected and reported in Table 1.

For fault diagnosis purpose, the complete model of the WT benchmark can be described as a nonlinear continuous-time dynamic model represented by the function \mathbf{f}_{wt} of Eq. (1) including the overall behaviour of the WT process reported in Fig. 1 with state vector \mathbf{x}_{wt} and fed by the driving input vector \mathbf{u} :

$$\begin{cases} \dot{\mathbf{x}}_{wt}(t) = \mathbf{f}_{wt}(\mathbf{x}_{wt}, \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{x}_{wt}(t) \end{cases} \quad (1)$$

Equation (1) highlights that the simulator allows to measure all the state vector signals, *i.e.* the rotor speed, the generator speed and the generated power of the WT process:

$$\mathbf{x}_{wt}(t) = \mathbf{y}(t) = [\omega_{g,m1}, \omega_{g,m2}, \omega_{r,m1}, \omega_{r,m2}, P_{g,m}]$$

The driving input vector is represented by the following signals:

$$\mathbf{u}(t) = [\beta_{1,m1}, \beta_{1,m2}, \beta_{2,m1}, \beta_{2,m2}, \beta_{3,m1}, \beta_{3,m2}, \tau_{g,m}]$$

that represent the acquired measurements of the pitch angles from the three WT blades and the measured generator/converter torque. These signals are acquired with sample time T in order to obtain N data indicated as $\mathbf{u}(k)$ and $\mathbf{y}(k)$ with index $k = 1, \dots, N$ that are exploited to design the fault diagnosis strategies addressed in this work.

3 Data–Driven Methods for Fault Diagnosis

This section recalls the fault diagnosis strategy proposed in this paper that relies on FS and NN tools, as summarised in Sect. 3.1. These architectures are able to represent NARX models exploited for estimating the nonlinear dynamic relations between the input and output measurements of the WT process and the fault signals. In this sense, these NARX prototypes will be employed as fault estimators for solving the problem of the fault diagnosis of the WT system.

Under these assumptions, the fault estimators derived by means of a data–driven approach represent the residual generators $\mathbf{r}(k)$, which provide the online reconstruction $\hat{\mathbf{f}}(k)$ of the fault signals summarised in Table tab:faults, as represented by Eq. (2):

$$\mathbf{r}(k) = \hat{\mathbf{f}}(k) \quad (2)$$

where the term $\hat{\mathbf{f}}(k)$ represents the general fault vector of Table tab:faults, i.e. $\hat{\mathbf{f}}(k) = \{\hat{f}_1(k), \dots, \hat{f}_9(k)\}$.

The fault diagnosis scheme exploiting the proposed fault estimators as residual generator is sketched in Fig. 2. Note that, as already highlighted, this scheme is also able to solve the fault isolation task [2].

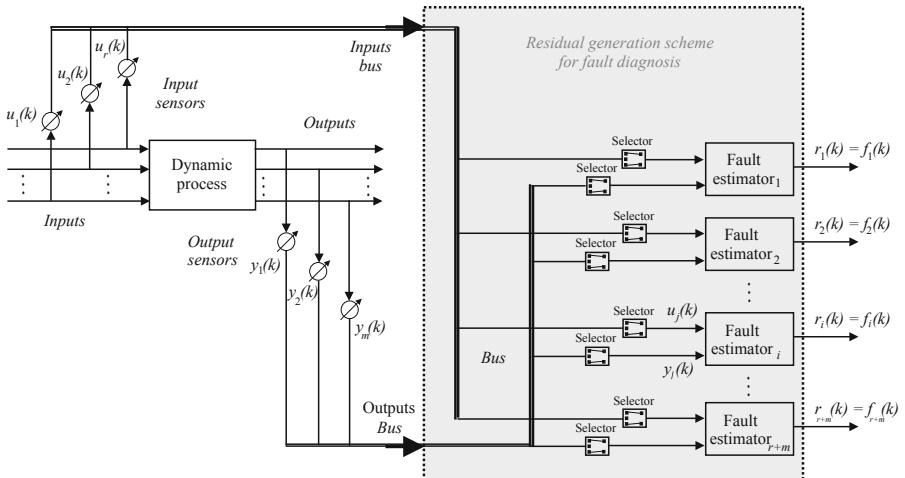


Fig. 2. Bank of fault reconstructors for fault diagnosis.

Figure 2 shows that the general residual generator exploits the input and output measurements acquired from the process under diagnosis, $\mathbf{u}(k)$ and $\mathbf{y}(k)$, properly selected according to the analysis shown in Table 1. The fault detection problem can be easily achieved by means of a simple threshold logic applied to the residuals themselves, as described in [2]. This issue will not be considered in this paper.

Once the fault detection phase is solved, the fault isolation stage is directly obtained via the bank of estimators of Fig. 2. In this case, the number of estimators of Fig. 2 is equal to the faults to be detected, *i.e.* 9, which is lower than the number of input and output measurements, $r + m$, acquired from the WT process.

This condition provides several degrees of freedom, as the i -th reconstructor of the fault $\hat{f}(k) = r_i(k)$ is a function of the input and output signals $\mathbf{u}(k)$ and $\mathbf{y}(k)$. These signals are thus selected in order to be affected sensitive to the specific fault $f_i(k)$, as highlighted in Table 1. This procedure enhances also the design of the fault reconstructors, as it reduces the number of possible input and output measurements, $u_j(k)$ and $y_l(k)$, which have to be considered for the identification procedure reported in Sect. 3.1.

The sensitivity analysis already represented in Table 1 has to be performed before the estimation of the fault estimators. Therefore, once the input–output signals are selected, according to Table 1, the FSs and the NNs used as fault reconstructors can be developed, as summarised in Sect. 3.1.

3.1 Fault Estimators via Artificial Intelligence Tools

This section recalls the procedure for developing the fault estimators modelled as Takagi–Sugeno (TS) FSs. In this way, the unknown dynamic relations between the selected input and output measurements of the WT plant and the faults are represented by means of FSs, which rely on a number of rules, antecedent and consequent functions. These rules are used to represent the inference system for connecting the measured signals from the system under diagnosis to its faults, in form of IF \Rightarrow THEN relations, implemented via the so-called Fuzzy Inference System (FIS) [1].

According to this modelling strategy, the general TS fuzzy prototype has the of Eq. (3):

$$\hat{f}(k) = \frac{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x}(k)) (\mathbf{a}_i^T \mathbf{x}(k) + b_i)}{\sum_{i=1}^{n_C} \lambda_i(\mathbf{x}(k))} \quad (3)$$

Using this approach, in general, the fault signal $\hat{f}(k)$ is reconstructed by using suitable data taken from the WT process under diagnosis. In this case, the fault function $\hat{f}(k)$ is represented as a weighted average of affine parametric relations $\mathbf{a}_i^T \mathbf{x}(k) + b_i$ (consequents) depending on the input and output measurements collected in $\mathbf{x}(k)$. These weights are the fuzzy membership degrees $\lambda_i(\mathbf{x})$ of the system inputs.

The parametric relations of the consequents depend on the unknown variables \mathbf{a}_i and b_i , which are estimated by means on an identification approach. The rule

number is assumed equal to the cluster number n_C exploited to partition the data via a clustering algorithm with respect to regions where the parametric relations (consequents) hold [1].

Note that the system under diagnosis corresponds to a WT plant, which is described by a dynamic model. Therefore, the vector $\mathbf{x}(k)$ in Eq. (3) contains both the current and the delayed samples of the system input and output measurements. Therefore, the consequents includes discrete-time linear Auto-Regressive with eXogenous (ARX) input structures of order o . This regressor vector is described in form of Eq. (4):

$$\mathbf{x}(k) = [\dots, y_l(k-1), \dots, y_l(k-o), \dots, u_j(k), \dots, u_j(k-o), \dots]^T \quad (4)$$

where $u_l(\cdot)$ and $y_j(\cdot)$ represent the l -th and j -th components of the actual WT input and output vectors $\mathbf{u}(k)$ and $\mathbf{y}(k)$. These components are selected according to the results reported in Table 1.

The consequent affine parameters of the i -th model of the Eq. (3) are usually represented with a vector:

$$\mathbf{a}_i = [\alpha_1^{(i)}, \dots, \alpha_o^{(i)}, \delta_1^{(i)}, \dots, \delta_o^{(i)}]^T \quad (5)$$

where usually the coefficients $\alpha_j^{(i)}$ are associated to the delayed output samples, whilst $\delta_j^{(i)}$ to the input ones.

The approach proposed in this paper for the derivation of the generic i -th fault approximator (FIS) starts with the fuzzy clustering of the data $\mathbf{u}(k)$ and $\mathbf{y}(k)$ from the WT process. This paper exploits the well-established Gustafson–Kessel (GK) algorithm [1]. Moreover, the estimation of the FIS parameters is addressed as a system identification problem from the noisy data of the WT process. Once the data are clustered, the identification strategy proposed in this work exploits the methodology developed by the authors in [3].

Another key point not addressed in this work concerns the selection of the optimal clusters number n_C . This issue was investigated and developed by the authors, which leads to the estimation of the membership degrees $\lambda_i(\mathbf{x}(k))$ required in Eq. (3) and solved as a curve fitting problem [1].

This paper considers an alternative data-driven approach, which exploits neural networks used as fault approximators in the scheme of Fig. 2. Therefore, in the same way of the fuzzy scheme, the bank of NNs is exploited to reconstruct the faults affecting the WT system under diagnosis using a proper selection of the input and the output measurements. The exploited NN structure consists of a feed-forward Multi-Layer Perceptron (MLP) architecture with 3 layers of neurons [5].

However, as MLP networks represent static relations, the paper suggests to implement the MLP structure with a tapped delay line. Therefore, this quasi-static NN represents a powerful way for estimating nonlinear dynamic regressions between the input and output measurements from the WT process and its fault functions. This solution allows to obtain another Nonlinear ARX (NARX)

description among the data. Moreover, when properly trained, these NARX NNs are able to reconstruct the fault function $\hat{f}(k)$ using a suitable selection of the past measurements of the WP system inputs and outputs $u_l(k)$ and $y_j(k)$, respectively. The example of the general solution is sketched in Fig. 3.

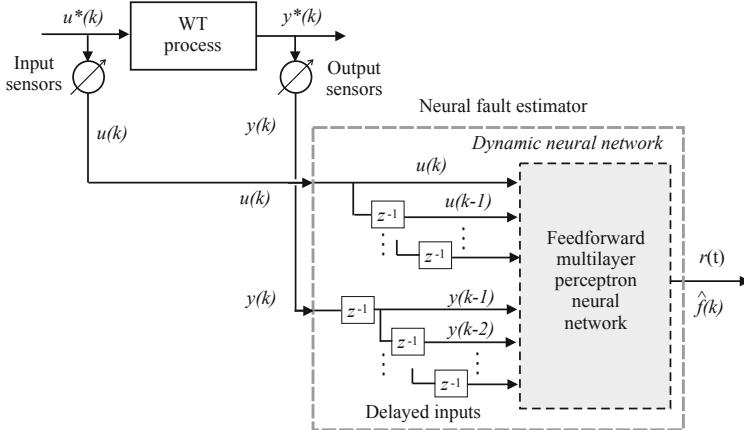


Fig. 3. NARX NN for fault reconstruction.

Similarly to the fuzzy scheme, with reference to the i -th fault reconstructor, a bank of NARX NNs is exploited, where the generic NARX system models the relation of Eq. (6):

$$\hat{f}(k) = F(\dots, u_j(k), \dots, u_j(k - d_u), \dots, y_l(k - 1), \dots, y_l(k - d_y), \dots) \quad (6)$$

where $\hat{f}(k)$ represents the estimate of the general i -th fault in Table 1, whilst $u_j(\cdot)$ and $y_l(\cdot)$ indicate the components of the measured inputs and outputs of the WT process. These signals are selected again by means of the solution of the fault sensitivity problem reported in Table 1. The accuracy of the fault reconstruction depends on the number of neurons per layer, their weights and their activation functions.

4 Simulation and Experimental Tests

This section reports the results of the simulations when the fault diagnosis schemes described in Sect. 3 are implemented for the WT process in Sect. 2. As already remarked, the WT benchmark implements realistic uncertainty and disturbance effects, which are considered for analysing the robustness characteristics of the designed fault diagnosis strategies. The same techniques are validated also using a more realistic HIL tool, as described in Sect. 4.2.

4.1 Simulated Results

With reference to the WT benchmark of Sect. 2, the simulations are driven by different wind sequences generated in a random way. They represent real measurements of wind speed sequences representing typical WT operating conditions, with ranges varying from 5 m/s. to 20 m/s. This scenario was modified by the authors with respect to the earlier benchmark proposed in [8]. The simulations consist of 4400 s., with single fault occurrences and a number of samples $N = 440000$ for a sampling frequency 100 Hz. Almost all fault signals are modelled as step functions lasting for 100 s. with different commencing times. Further details can be found in [8, 10].

The first part of this section reports the results achieved by means of the fuzzy prototypes used as fault reconstructors according to Sect. 3.1. In particular, the fuzzy c -means and the GK clustering algorithms were exploited. A number of clusters $n_C = 4$ of clusters and a number of delays $o = 3$ were estimated. The membership functions of the TS FS and the parameters of the consequents $\alpha_j^{(i)}$ and $\delta_j^{(i)}$ were estimated for each cluster by following the procedure developed by the same authors in [12]. The TS FSs of Eq. (3) were thus determined and 9 fault reconstructors were organised according to the scheme of Fig. 2.

The performances of the 9 TS FSs when used as fault estimators were evaluated again according to the RMSE % index, computed as the difference between the reconstructed $\hat{f}(k)$ and the actual $f(k)$ signals for each of the fuzzy estimators. These values were reported in Table 2.

Table 2. FS fault estimator capabilities.

Fault case	1	2	3	4	5	6	7	8	9
RMSE%	1.61%	2.22%	1.95%	1.87%	1.92%	2.15%	1.76%	2.13%	1.98%
Sdt. Dev.	$\pm 0.02\%$	$\pm 0.03\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.02\%$	$\pm 0.01\%$	$\pm 0.02\%$	$\pm 0.01\%$

Indeed, the RMSE % values reported in Table 2 represent an average of the results obtained from a campaign of 1000 simulations, as the benchmark exploited in this work changes the parameters of the WT model at each run. Moreover, the model–reality mismatch, the measurement errors, uncertainty and disturbance effects are described as Gaussian processes with suitable distributions, as remarked in Sect. 2. Therefore, Table 2 reports also the values of the standard deviation of the estimation errors achieved by the FS fault estimators.

Note that these reconstructed signals $\hat{f}(k)$ can be directly used as diagnostic residuals in order to detect and isolate the faults affecting the WT. Moreover, each TS FS of Eq. (3) is fed by 3 inputs (according to Table 1), with a number of delayed inputs and outputs $n = 3$ and $n_C = 4$ clusters.

As an example, Fig. 4 shows the results regarding the fault cases 1, 2, 3, and 4 of the WT plant recalled in Sect. 2.

In particular, Fig. 4 reports the estimated faults $\hat{f}(k) = r_i(k)$ provided from the FSs in faulty conditions (black continuous line). They are compared with

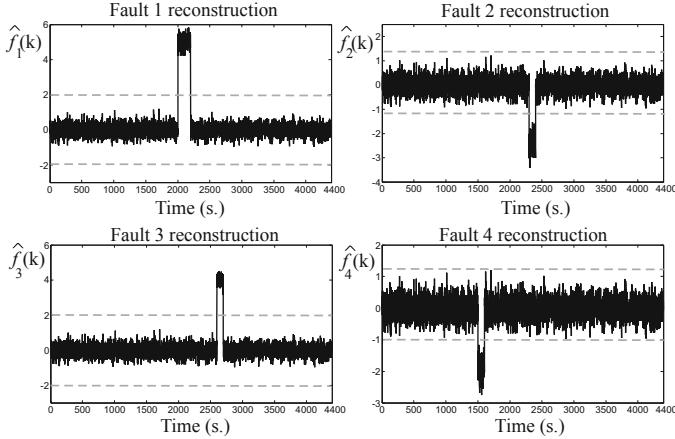


Fig. 4. Reconstructed faults $\hat{f}(k)$ for cases 1, 2, 3, and 4.

respect to the corresponding fault-free residuals (grey line). The fixed thresholds reported with dotted lines are used for fault detection. Note that the reconstructed fault functions $\hat{f}(k) = r_i(k)$ are different from zero also in fault-free conditions due to the measurement errors and the model-reality mismatch. This aspect serves to highlight the accuracy of the reconstructed signals provided by the estimated fuzzy models.

As for the FSs, 9 NARX NNs summarised in Sect. 3.1 were derived to provide the reconstruction of the 9 faults affecting the WT plant. In particular, the NARX were implemented as MLP NNs with 3 layers: the input layer consisted of 3 neurons, the hidden one used 10 neurons, whilst one neuron for the output layer. 4 delays were used in the relation of Eq. (6). Moreover, sigmoidal activation functions were used in both the input and the hidden layers, and a linear function for the output layer. With reference to Table 1, the NARX NNs were fed by 9 signals, representing the delayed inputs and outputs from the WT process.

As for the FSs, the prediction accuracy of the NARX NN was analysed by means of the RMSE % index and its average values summarised in Table 3.

Table 3. NN fault estimator capabilities.

Fault case	1	2	3	4	5	6	7	8	9
RMSE %	0.91%	0.92%	0.94%	1.21%	1.17%	1.61%	0.98%	0.95%	1.41%
Sdt. Dev.	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.02\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.01\%$	$\pm 0.02\%$

As for the FS case, Table 3 reports also the values of the standard deviation of the estimation errors achieved by the NARX NN fault estimators.

Also in this case, Fig. 5 depicts some of the residual signals $\hat{f}(k) = r_i(k)$ provided by the NARX NNs for the fault conditions 6, 7, 8, and 9, and compared with respect to the fixed detection thresholds (dotted lines).

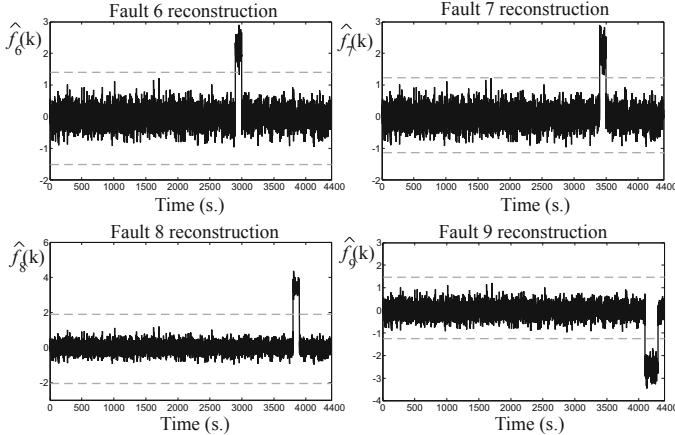


Fig. 5. Estimated faults for cases 6, 7, 8, and 9.

Also in this case, the results obtained by the NARX NNs serve to highlight the efficacy of the developed solution, taking into account also disturbance and uncertainty affecting the WT system.

4.2 Hardware-in-the-Loop Experiments

In order to validate the developed fault diagnosis solutions in more realistic real-time working situations, the WT process and the designed algorithms have been implemented and executed by means of a HIL tool. This test-bed allows to reproduce experimental tests that are oriented to the verification of the results achieved in simulations. This test-bed is sketched in Fig. 6, which highlights its 3 main modules.

The WT simulator developed in the Matlab and Simulink environments that was used to describe WT system dynamics, its actuator, measurement sensors, and the WT controlled has been implemented in the LabVIEW environment. Realistic effects such as uncertainty, measurement errors, disturbance and the model-reality mismatch effects were also included, as recalled Sect. 2. The overall system is converted in the C++ code running on a standard PC, and allows also to test and monitor the signals generated by the proposed fault diagnosis strategies.

These fault diagnosis schemes summarised in Sect. 3.1 have been also compiled as executable code and implemented in an AWC 500 industrial system that features typical wind turbines requirements. This industrial module receives the

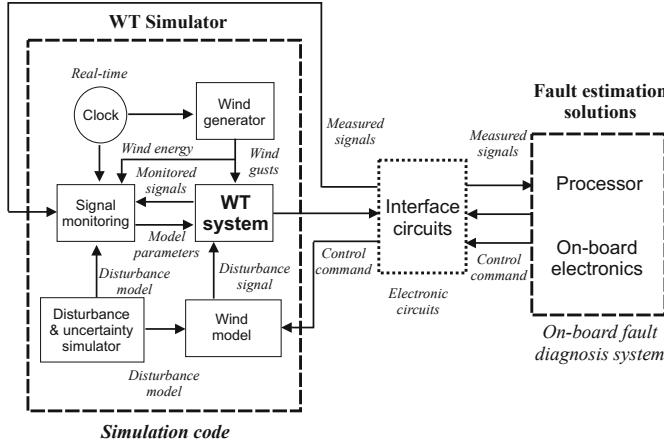


Fig. 6. HIL tool for real-time validation.

signals acquired from the PC simulating the realistic WT plant that represent the monitored signals reported in Table 1. Therefore, the on board electronics elaborate these signals according to the fault diagnosis algorithms and produce the monitoring signals transmitted back to the WT simulator running on the PC.

An intermediate module represents the interface circuits providing the communications between the PC with the WT simulator and the on board electronics running the fault diagnosis algorithms. In this way, it manages the signals and exchanges the data between the WT simulator and the AWC 500 system.

The results achieved via this HIL tool are reported in Table 4 that summarises the capabilities of the fault diagnosis algorithms by means of the NSSE % performance index.

Table 4. RMSE % index for the HIL tool.

Fault case	1	2	3	4	5	6	7	8	9
TS FSs	1.69%	2.29%	2.01%	1.94%	1.99%	2.22%	1.81%	2.21%	2.03%
NARX NNs	0.99%	0.98%	0.99%	1.28%	1.21%	1.69%	1.02%	1.01%	1.51%

Note that the tests summarised in Table 4 are consistent with the results reported in Tables 2 and 3. Although the accuracy of the simulations seems better than the performance achieved via the HIL tool, some remarks have to be drawn. First, the AWC 500 system uses calculations that are more restrictive than the PC simulator. Moreover, A/D and D/A devices are also exploited, which can introduce further deviations. On the other hand, the testing of real scenarios does not involve the data transfer from a PC to on board electronics,

thus reducing possible errors. Therefore, it can be finally remarked that the achieved results are quite accurate and motivate the application of the developed fault diagnosis strategies to real WT installations.

5 Conclusion

The paper addressed fault diagnosis strategies that were developed by means of artificial intelligence tools. In this way, the challenging problem of the condition monitoring of a wind turbine process was solved via data–driven approaches. The design fault diagnosis strategy was based on the reconstruction of the faults affecting the considered process, thus implemented via dynamic system identification and artificial intelligence tools. The paper proposed these strategies as they represented viable methods that were able to manage partial known information on the process dynamics, and to cope with measurement errors, model–reality mismatch and disturbance effects. In particular, these fault diagnosis schemes were implemented by means of fuzzy and neural network structures exploited determine the nonlinear dynamic relations between the input and output measurements acquired from the process under diagnosis and the fault signals affecting the process. These developed prototypes included auto–regressive with exogenous input architectures in order to approximate the nonlinear dynamic relations with arbitrary degree of accuracy. The designed fault diagnosis solutions were validated via a high–fidelity benchmark that simulated the behaviour of a realistic wind turbine plant. This wind turbine simulator was also employed to test the reliability and robustness characteristics of the fault diagnosis schemes by considering the presence of uncertainty and disturbance effects included in this wind turbine benchmark. Further works will verify the features of the same fault diagnosis schemes when applied to real plants and by means of real data.

References

1. Babuška, R.: *Fuzzy Modeling for Control*. Kluwer Academic Publishers, Boston (1998)
2. Chen, J., Patton, R.J.: *Robust Model-Based Fault Diagnosis for Dynamic Systems*. Kluwer Academic Publishers, Boston (1999)
3. Fantuzzi, C., Simani, S., Beghelli, S., Rovatti, R.: Identification of piecewise affine models in noisy environment. *Int. J. Control.* **75**(18), 1472–1485 (2002). <https://doi.org/10.1109/87.865858>
4. Harrabi, N., Kharrat, M., Aitouche, A., Souissi, M.: Control strategies for the grid side converter in a wind generation system based on a fuzzy approach. *Int. J. Appl. Math. Comput. Sci.* **28**(2), 323–333 (2018)
5. Korbicz, J., Koscielny, J.M., Kowalcuk, Z., Cholewa, W. (eds.): *Fault Diagnosis: Models, Artificial Intelligence, Applications*, 1st edn. Springer, London, 12 February 2004. ISBN 3540407677
6. Lan, J., Patton, R.J., Zhu, X.: Fault-tolerant wind turbine pitch control using adaptive sliding mode estimation. *Renew. Energy* **116**(Part B), 219–231 (2018). <https://doi.org/10.1016/j.renene.2016.12.005>

7. Odgaard, P.F., Shafiei, S.E.: Evaluation of wind farm controller based fault detection and isolation. In: Proceedings of the IFAC SAFEPROCESS Symposium 2015, Paris, France, September 2015, vol. 48, pp. 1084–1089. IFAC, Elsevier (2015). <https://doi.org/10.1016/j.ifacol.2015.09.671>
8. Odgaard, P.F., Stoustrup, J., Kinnaert, M.: Fault-tolerant control of wind turbines: a benchmark model. *IEEE Trans. Control Syst. Technol.* **21**(4), 1168–1182 (2013). <https://doi.org/10.1109/TCST.2013.2259235>. ISSN 1063-6536
9. Odgaard, P.F., Stoustrup, J.: A benchmark evaluation of fault tolerant wind turbine control concepts. *IEEE Trans. Control Syst. Technol.* **23**(3), 1221–1228 (2015)
10. Odgaard, P.F., Stoustrup, J., Kinnaert, M.: Fault tolerant control of wind turbines - a benchmark model. In: Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Barcelona, Spain, 30 June–3 July 2009, vol. 1, pp. 155–160 (2009). <https://doi.org/10.3182/20090630-4-ES-2003.0090>
11. Parker, M.A., Ng, C., Ran, L.: Fault-tolerant control for a modular generator-converter scheme for direct-drive wind turbines. *IEEE Trans. Ind. Electron.* **58**(1), 305–315 (2011)
12. Rovatti, R., Fantuzzi, C., Simani, S.: High-speed DSP-based implementation of piecewise-affine and piecewise-quadratic fuzzy systems. *Signal Process. J.* **80**(6), 951–963 (2000). Special Issue on Fuzzy Logic applied to Signal Processing. [https://doi.org/10.1016/S0165-1684\(00\)00013-X](https://doi.org/10.1016/S0165-1684(00)00013-X)
13. Simani, S.: Application of a data-driven fuzzy control design to a wind turbine benchmark model. In: Advances in Fuzzy Systems 2012, 2nd November 2012, Invited paper for the special issue: Fuzzy Logic Applications in Control Theory and Systems Biology (FLACE), pp. 1–12 (2012). ISSN 1687-7101, e-ISSN 1687-711X. <https://doi.org/10.1155/2012/504368>. <http://www.hindawi.com/journals/afs/2012/504368/>
14. Simani, S., Castaldi, P.: Data-driven design of fuzzy logic fault tolerant control for a wind turbine benchmark. In: Astorga-Zaragoza, C.M., Molina, A. (eds.) 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes - SAFEPROCESS 2012, Mexico City, Mexico, 29th–31st August 2012, vol. 8, pp. 108–113. Instituto de Ingeniería, Circuito escolar, Ciudad Universitaria, CP 04510, México D.F., IFAC. Invited session paper (2012). ISBN 978-3-902823-09-0. ISSN 1474-6670. <https://doi.org/10.3182/20120829-3-MX-2028.00036>
15. Simani, S., Fantuzzi, C., Rovatti, R., Beghelli, S.: Parameter identification for piecewise linear fuzzy models in noisy environment. *Int. J. Approx. Reason.* **1**(22), 149–167 (1999)
16. Simani, S., Farsoni, S., Castaldi, P.: Active fault tolerant control of wind turbines using identified nonlinear filters. In: Proceedings of the 2nd International Conference on Control and Fault-Tolerant Systems - SysTol 2013, Nice, France, 9–11 October 2013, pp. 383–388. Centre de Recherche en Automatique de Nancy - CRAN. IEEE Control Systems Society (2013). Special session invited paper ISBN 978-1-4799-2854-5. <https://doi.org/10.1109/SysTol.2013.6693827>
17. Simani, S., Farsoni, S., Castaldi, P.: Fault tolerant control of an offshore wind turbine model via identified fuzzy prototypes. In: Whidborne, J.F. (ed.) Proceedings of the 2014 UKACC International Conference on Control (CONTROL), Loughborough University, Loughborough, UK, 8th–11th July 2014, pp. 494–499. UKACC (United Kingdom Automatic Control Council), IEEE (2014). ISBN 9781467306874. Special session invited paper. <https://doi.org/10.1109/CONTROL.2014.6915188>

18. Simani, S., Farsoni, S., Castaldi, P.: Residual generator fuzzy identification for wind farm fault diagnosis. In: Proceedings of the 19th World Congress of the International Federation of Automatic Control - IFAC 2014, Cape Town, South Africa, 24–29 August 2014, vol. 19, pp. 4310–4315. IFAC & South Africa Council for Automation and Control, IFAC (2014). Invited paper for the special session “FDI and FTC of Wind Turbines in Wind Farms” organised by P. F. Odgaard and S. Simani. <https://doi.org/10.3182/20140824-6-ZA-1003.00052>
19. Simani, S., Farsoni, S., Castaldi, P.: Data-driven techniques for the fault diagnosis of a wind turbine benchmark. *Int. J. Appl. Math. Comput. Sci.* - AMCS **28**(2), 247–268 (2018). <https://doi.org/10.2478/amcs-2018-0018>
20. Xu, F., Puig, V., Ocampo-Martinez, C., Olaru, S., Niculescu, S.I.: Robust MPC for actuator-fault tolerance using set-based passive fault detection and active fault isolation. *Int. J. Appl. Math. Comput. Sci.* **27**(1), 43–61 (2017). <https://doi.org/10.1515/amcs-2017-0004>



Mathematical Model to Support Decision-Making to Ensure the Efficiency and Stability of Economic Development of the Republic of Kazakhstan

Askar Boranbayev^{1(✉)}, Seilkhan Boranbayev², Malik Baimukhamedov³, and Askar Nurbekov²

¹ Nazarbayev University, Nur-Sultan, Kazakhstan

aboranbayev@nu.edu.kz

² L.N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan

³ Kostanay Socio-Technical University named after Academician

Z. Aldamzhar, Kostanay, Kazakhstan

Abstract. At planning the state program and the regional programs of economic development it is necessary to define the total of resources, which are required for performance of these programs. It is required to calculate performance of programs at the set plan of deliveries of resources. Such calculations can be carried out on models of forward planning of development of economy and mathematical models of process of performance of economic programs. In the article the mathematical model and a method of calculation of process of performance of economic programs is resulted. The mathematical model is necessary for a choice of different variants of programs. The suggested mathematical model enables to build the schedule of performance of a complex of programs in directive terms at the set plan receipt of resources, or to be convinced that the given complex of programs cannot be executed in directive terms. The analyzed variants of programs are considered set. The program is considered set, if the set of operation, which should be executed, and resources, which are necessary for performance of each of operation, are set. For the mathematical description of a complex of programs the network is used.

Keywords: Operations · Resources · Network · Schedule · Management

1 Introduction

The Republic of Kazakhstan is administratively divided into 14 regions. Each of the regions has its own planning and management bodies, which are an integral part of the state planning and management body. Regional economic development programs should be planned taking into account the programs of other regions and the general state program. Each of the regions separately does not produce all types of resources necessary for the implementation of the state development program. Therefore, ensuring

the efficiency and stability of the economic development of the Republic of Kazakhstan depends on the optimal distribution of resources in the regions.

The goal of state bodies is to develop and maintain the stability of the state. This goal is the source of the formation of the state development program, which, in turn, may consist of several programs. The following main ways of maintaining the stability of the state can be distinguished: meeting the needs of the regions; implementation of national programs, such as defense programs, environmental protection programs and others. Thus, the state and the regions have common goals. They are the source of the formation of parts of national programs, which take into account the needs of the regions. Thus, planning the development of the region cannot be viewed as an isolated task. It should be considered in conjunction with planning the development of the entire state system. Regional development planning procedures are part of the state development planning procedures. Many methods are devoted to solving problems of optimizing economic development, in particular [1–10].

We will assume that most of the resources are subordinate to government bodies or are under double subordination. In particular, the natural resources located on the territory of the region are at the disposal of the state. At the same time, the population living in the territory of this region is the source of the formation of labor resources. Therefore, the region has an important type of resources - labor resources. The labor resources of the region are used in all types of property and in all funds (state, regional, mixed, foreign and others). They are not directly at the disposal of state and regional authorities. The structure and quality of labor resources, as well as the stability of the region and the state system as a whole, depend on the degree of satisfaction of the needs of the regions. It can be considered that the regional planning and management bodies should determine the needs of the population of the region and the degree of their satisfaction. Thus, the quantity and quality of labor resources depends on the degree to which the needs of the region's population are met.

The procedures for the development of state and regional development programs should include methods for assessing the total amount of resources that can be allocated for the implementation of programs and the calculation of program implementation for a given resource supply plan. This kind of calculations can be carried out on models of long-term planning of the economy and models of the process of implementation of economic programs.

This paper provides a mathematical model and a method for calculating the process of implementing economic programs. The model is needed to select different program options. It makes it possible to build a schedule for the execution of programs on target dates for a given resource flow, or to make sure that a given program cannot be completed on target dates.

2 Problem Statement and Its Mathematical Model

Let's consider a directed network K , top i of a network we shall designate K_i . Vertices of network designate some operations which should be executed. Arches of a network mean technological connections between these operations. Directions of the arches which are starting on top K_i , specify tops which states are influenced with a state of top K_i .

Set of all operation of a network we shall designate $I, I = \{1, 2, \dots, n\}$. Each operation we shall attribute conditional number, some integer $i, i \in I$. We shall designate I_i^+ operation set, directly previous to operation i , that is operation i cannot start before all operations from set $I_i^+, I_i^+ \subset I$ will end. Thus, in I_i^+ enter those operations which directly influence a condition of operation i . We shall designate I_i^- operation set, directly following for operation i , that is if operation i is not finished, operations from set I_i^- cannot start to be carried out. Thus, operation i directly influences on conditions of operation from set I_i^- . We assume, that $i \in I_i^+, i \in I_i^-$, that is the network does not contain loops.

Each operation $i \in I$ demand for execution a resources. Set of kinds of resources we shall designate $J, J = \{1, 2, 3, \dots, m\}$.

I_j - operation set, using a resource of the j kind. We shall designate $x_i(t)$ a condition of operation i , also, we shall name $x_i(t)$ a number of operation i . When the number $x_i(t)$ of operation i will reach value of unity, operation is considered executed. At $x_i(t) = 0$ performance of operation i did not begin yet, at $0 < x_i(t) < 1$ operation is executed partly.

The magnitude $x_i(t)$ is determined for each operation i in own way, depending on the concrete contents of operation. For example, it can be time of performance of operation or volume of operation in some units, etc. Thus, the condition of all complex of operation of a network can be described a vector $x = (x_1, x_2, \dots, x_n)$. Value x varies eventually. The moment of time, in which vector x for the first time coincides with unit vector, will be the moment of end of all complex of operation.

Intensity of performance of operation in given moment of time we shall name speed of increase of a number of a condition of operation during this moment of time.

Operation uses resources during the performance and returns it back after the end. The resources which have been not used during some moment of time, do not increase amount of resources during the subsequent moments of time, they do not accumulate.

Let's designate u_i a condition of managing objects for operation $i, i \in I$. Managements, in a problem of calculation of performance in a complex of a network's operation, are intensity of the operation, appointed for performance at present time. We shall designate u as intensity performance vector of a complex of operation at present time, $u = (u_1, u_2, \dots, u_n)$, where u_i - intensity performance of operation i at present time. The condition of operation i depends on a condition of operation $v, v \in I_i^+ \subset I$, intensity u_i , and also is defined by value of the parameters.

Process of performance of a complex of operation of a network is controlled. There is certain freedom in operation select which will be carried out at present time, and intensity their performance. The problem consists in that for each moment of time, at existing restrictions on resources, should be named a vector of intensity u with which it is necessary for operation of a network to carry out. As the best are considered such u which deliver a minimum to the some functional.

All restrictions with which should satisfy intensity u , are broken into restrictions of two types.

Restrictions of the first type describe interdependence of performance of operation (the attitude of precedence). Generally restrictions of the first type are described by some function which to each condition x of a complex of operation of a network puts

in conformity allowable values u operations intensity, from some set $U(x)$ with which operations of a complex in the given condition can be carried out. Thus, generally, restrictions of the first type look like:

$$u \in U(x).$$

Restrictions of the second type include restrictions on resources. Generally the vector of the resources spent on performance of a complex of operation at the moment of time t , depends on conditions $x(t)$ of a complex of operations and intensity $u(t)$ at the moment of time t . We shall designate $V(t)$ - a vector of the total resources available in system at the moment of time t . $D(x(t), u(t))$ - total of the resources spent on performance of operation of a complex during the moment t . Then, generally, restrictions of the second type look like:

$$D(x(t), u(t)) \leq V(t).$$

If the vector $V(t)$ during each moment of time is set, calculation of performance of a complex of operations is reduced to the following: for each moment t the vector intensity $u(t)$ with which it is necessary to carry out operations of a complex should be named. It is considered the best such $u(t)$ at which some функционал will be minimal. As functional it is possible to take, for example, a degree of satisfiability of complex of operation by some directive fixed moment of time T .

Let's take functional in the following kind:

$$F = \frac{1}{1+\alpha} \sum_{i=1}^n \lambda_i (1 - x_i(T))^{1+\alpha}$$

Where $\alpha \geq 0 \lambda_i \geq 0$.

It is possible to replace restrictions of the first type the penalty for their infringement. As is known, computing methods of search of the optimum decision can be treated as use of penalties. In one cases the penalty is imposed without taking into account features of a problem, in other cases there is a communication between a deviation from the optimum decision and size of the penalty.

The penalty for infringement of restriction of the first type we shall take into account in functional. Then instead of functional F , it is optimized functional with the penalty:

$$\Phi = F + \int_0^T \varphi(x, u, t, \varepsilon) dt \quad (1)$$

Where parameter $\varepsilon > 0$.

The second composed in the right part (1) increases value functional if restrictions of the first type are broken. If these conditions are executed, the second composed in the right part (1) is equal to zero and values functional Φ and F coincides. Hence, their minimal values are realized by the same function of management.

For definiteness function $\varphi(x, u, t, \varepsilon)$ undertakes in the following kind:

$$\varphi(x, u, t, \varepsilon) = \frac{1}{\varepsilon} \sum_{i \in I} u_i(t) \sum_{k \in I_i^+} \theta(1 - x_k(t))$$

$$\text{Where } \theta(\tau) = \begin{cases} 0, & \tau < 0 \\ 1, & \tau > 0 \\ 0, & \tau = 0 \end{cases}$$

Then the problem will look like:

$$\begin{aligned} \Phi = & \frac{1}{1+\alpha} \sum_{i=1}^n \lambda_i (1 - x_i(T))^{1+\alpha} \\ & + \frac{1}{\varepsilon} \int_0^T \sum_{i \in I} u_i(t) \sum_{k \in I_i^+} \theta(1 - x_k(t)) dt \rightarrow \min \\ & \frac{dx_i(t)}{dt} = u_i(t), \quad i \in I \end{aligned}$$

$$x_i(0) = 0, \quad i \in I$$

$$x_i(T) = 1, \quad i \in I$$

$$\sum_{i \in I_j} d_{ji}(t) u_i(t) \leq V_j(t), \quad j \in J, \quad V_j(t) > 0$$

$$0 \leq u_i(t) \leq H_i(t), \quad i \in I$$

$$\forall i \exists j : d_{ji}(t) > 0, \quad j \in J.$$

Where $d_{ji}(t)$ - intensity of an expenditure resource j at performance operations i with individual intensity;

$V_j(t)$ - intensity of receipt of a resource j - th kind, step function.

It is necessary to finish all complex of operation for directive time T .

If restrictions of the first type are not executed,

$$u_i(t) \sum_{k \in I_i^+} \theta(1 - x_k(t)) > 0, \quad i \in I$$

If restrictions of the first type are executed,

$$u_i(t) \sum_{k \in I_i^+} \theta(1 - x_k(t)) = 0, \quad i \in I$$

and both functional Φ and F coincide.

3 Method and Algorithm for Solving the Problem

The problem under consideration is solved by the method proposed in the article [1]. Use of this method allows to carry out decomposition of an initial problem on sequence of problems of linear programming not the big dimension. Thus, the problem of planning and distribution of resources of a network with directive terms of performance of a complex of operation of a network is reduced to sequence of problems of construction of the decision for set technological independent operations on the basis of methods of local optimization.

Let's note, that at calculation of the networks containing a plenty of operation, there can be the certain difficulties connected to high dimension of a problem. In this case it is expedient to replace a problem of high dimension with several problems of small dimension. We shall consider the methods, allowing it to make.

- 1) The interval $[0, T]$ is broken into some intervals $[0, T_1], [T_1, T_2], \dots, [T_{k-1}, T_k], \dots$

On an interval $[0, T_1]$ resources are distributed not for all operations I of a network, but only for some operations I_1 subnet, $I_1 \subset I$. After on a method stated in clause [1] it is defined optimum on $[0, T_1]$ the decision, pass to the following interval of time $[T_1, T_2]$. On an interval $[T_1, T_2]$ are calculated $u_i(t)$ for operation from set I_2 , $I_2 \subset I$. The set I_2 will consist of the operations which are included in I_1 and whose performance has not ended till the moment T_1 , and some operations from I , not belonging I_1 .

After the decision is received on k interval, pass on $(k + 1)$ interval. Thus:

$$x_i^{k+1}(T_k) = x_i^k(T_k), \quad i \in I_k$$

$$x_i^{(k+1)}(T_k) = 0, \quad i \in I \setminus \bigcup_{s=1}^k I_s,$$

that is, the initial condition of operation on an interval $k + 1$ coincides with value of a condition of operation during the corresponding moment of time on an interval k .

After decisions on each such interval will be found, the basic iteration of algorithm comes to an end. If the complex of operation of a network will be completed in directive terms the required decision is found. If the complex of operation of a network will not be completed in directive terms the following iteration for new splitting a network on sub network is carried out, in view of the decision constructed on previous iteration.

Let's note, that calculation on the resulted method allows to receive decisions with the earliest moments of the beginning of performance of operation with the maximal surplus of resources at the moment of time T .

- 2) If the iterative process described above carry out, since last site it is possible to receive the schedule which is similar to the schedule of dynamic programming [2]. For this purpose we shall enter a variable:

$$\tau = T - t.$$

The interval $[0, T]$ is broken into subintervals $[0, \tau_1], [\tau_1, \tau_2], \dots, [\tau_k, \tau_{k+1}], \dots$

The method resulted in [1] and decision is applied to each of subintervals is defined. On the first interval $[0, \tau_1]$ the decision for sub network S_1 , $S_1 \subset R$, where R - a network, inverse to an initial network K . Then we find solution on the following interval $[\tau_1, \tau_2]$ for sub network S_2 , $S_2 \subset R$. Sub network S_2 will consist on operation of sub network S_1 , not reached a zero level of a point and the some operation from $R \setminus S_1$.

After the decision on interval k is received, pass on interval $(k + 1)$. Thus,

$$x_i^{k+1}(\tau_k) = x_i^k(\tau_k), \quad i \in S_k$$

$$x_i^{k+1}(\tau_k) = 1, \quad i \in R \setminus \bigcup_{r=1}^k S_r$$

Let's note, that this method allows to receive decision with the latest moments of time of the termination of performance of operation with the minimal surplus of resources at the moment of time T .

A software system has been developed that alternates calculations by these two methods and allows you to get the best solution. The data for the calculations are taken from the respective databases.

For uninterrupted and trouble-free operation of the control system, it is necessary to ensure the reliability and safety of automated information systems [11–41]. When developing a software system for modeling and supporting decision-making to ensure the efficiency and stability of economic development, technologies and methods were used to increase the level of their reliability and safety.

When developing this program, the following systems were used: Delphi, ADO technology, MS SQL Server database management system.

4 Conclusion

The development goals of the regions are determined by their needs, the degree of their satisfaction, and the system of legal relations. The mismatch between needs and the degree of their satisfaction is the source of the formation of regional programs of economic development.

We consider a state with centralized control, and the resources of the state are greater than the resources of the regions. National interests in decision-making procedures are the starting point for the planning of economic development.

The article developed a mathematical model and a calculation method for the selection of various options for economic development programs. The developed method makes it possible to build a schedule for the execution of programs in the target time frame for a given plan of resource inflow, or to make sure that the given program cannot be executed in the target time frame. To form a program of economic development, it is necessary to assign a set of operations to be performed and the resources that are required to complete each of the operations.

References

1. Boranbayev, S.N.: The algorithm of decision task of planning and distribution of net resources by methods of sequence approaches. In: Materials of International Science Practical Conference “Ualihanov’s Reading”, Kokshetau, pp. 191–194 (2003)
2. Bellman, R.E.: Dynamic Programming. Princeton University Press, Princeton (1957)
3. Hartman, J.: Engineering Economy and the Decision-Making Process, Upper Saddle River. Pearson Prentice Hall, Hoboken (2007)
4. Yatsenko, Y., Hritonenko, N.: Economic life replacement under improving technology. *Int. J. Prod. Econ.* **133**, 596–602 (2011)
5. Yatsenko, Y., Hritonenko, N.: Machine replacement under evolving deterministic and stochastic costs. *Int. J. Prod. Econ.* **193**, 491–501 (2017)
6. Yatsenko, Y., Hritonenko, N.: Asset replacement under improving operating and capital costs: a practical approach. *Int. J. Prod. Res.* **54**, 2922–2933 (2016)
7. Hritonenko, N., Yatsenko, Y., Boranbayev, S.: Non-equal-life asset replacement under evolving technology: a multi-cycle approach. *Eng. Econ.* **25** (2020)
8. Hritonenko, N., Yatsenko, Y., Boranbayev, S.: Environmentally sustainable industrial modernization and resource consumption: is the Hotelling’s rule too steep? *Appl. Math. Model.* **39**(15), 4365–4377 (2015)
9. Boranbayev, S.N., Nurbekov, A.B.: Construction of an optimal mathematical model of functioning of the manufacturing industry of the Republic of Kazakhstan. *J. Theor. Appl. Inf. Technol.* **80**(1), 61–74 (2015)
10. Hritonenko, N., Yatsenko, Y., Boranbayev, A.: Generalized functions in the qualitative study of heterogeneous populations. *Math. Popul. Stud.* **26**(3), 146–162 (2019)
11. Boranbayev, A., Boranbayev, S., Nurusheva, A., Yersakhanov, K.: Development of a software system to ensure the reliability and fault tolerance in information systems. *J. Eng. Appl. Sci.* **13**(23), 10080–10085 (2018)
12. Boranbayev, S., Goranin, N., Nurusheva, A.: The methods and technologies of reliability and security of information systems and information and communication infrastructures. *J. Theor. Appl. Inf. Technol.* **96**(18), 6172–6188 (2018)
13. Boranbayev, A., Boranbayev, S., Nurusheva, A.: Development of a software system to ensure the reliability and fault tolerance in information systems based on expert estimates. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) Intelligent Systems and Applications, vol. 869, pp. 924–935. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-01057-7_68
14. Boranbayev, A., Boranbayev, S., Yersakhanov, K., Nurusheva, A., Taberkhan, R.: Methods of ensuring the reliability and fault tolerance of information systems. In: Latifi, S. (ed.) Information Technology – New Generations. AISC, vol. 738, pp. 729–730. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77028-4_93
15. Boranbayev, S., Altayev, S., Boranbayev, A.: Applying the method of diverse redundancy in cloud based systems for increasing reliability. In: The 12th International Conference on Information Technology: New Generations (ITNG 2015), Las Vegas, Nevada, USA, 13–15 April 2015, pp. 796–799 (2015)
16. Turskis, Z., Goranin, N., Nurusheva, A., Boranbayev, S.: A fuzzy WASPAS-based approach to determine critical information infrastructures of EU sustainable development. *Sustainability* **11**(2), 424 (2019). <https://doi.org/10.3390/su11020424>
17. Turskis, Z., Goranin, N., Nurusheva, A., Boranbayev, S.: Information security risk assessment in critical infrastructure: a hybrid MCDM approach. *Informatica* **30**(1), 187–211 (2019). <https://doi.org/10.15388/Informatica.2018.203>
18. Boranbayev, A.S., Boranbayev, S.N., Nurusheva, A.M., Yersakhanov, K.B., Seitkulov, Y.N.: Development of web application for detection and mitigation of risks of information and automated systems. *Eurasian J. Math. Comput. Appl.* **7**(1), 4–22 (2019)

19. Boranbayev, A., Boranbayev, S., Nurusheva, A., Seitkulov, Y., Sissenov, N.: A method to determine the level of the information system fault-tolerance. *Eurasian J. Math. Comput. Appl.* **7**(3), 13–32 (2019). <https://doi.org/10.32523/2306-6172-2019-7-3-13-32>
20. Boranbayev, A., Boranbayev, S., Nurbekov, A., Taberkhan, R.: The development of a software system for solving the problem of data classification and data processing. In: Latifi, S. (ed.) 16th International Conference on Information Technology-New Generations (ITNG 2019). AISC, vol. 800, pp. 621–623. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14070-0_90
21. Askar, B., Seilkhan, B., Assel, N., Kuanysh, Y., Yerzhan, S.: A software system for risk management of information systems. In: Proceedings of the 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT 2018), Almaty, Kazakhstan, 17–19 October 2018, pp. 284–289 (2018)
22. Seilkhan, B., Askar, B., Sanzhar, A., Askar, N.: Mathematical model for optimal designing of reliable information systems. In: Proceedings of the 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT 2014), Astana, Kazakhstan, 15–17 October 2014, pp. 123–127 (2014)
23. Seilkhan, B., Sanzhar, A., Askar, B., Yerzhan, S.: Application of diversity method for reliability of cloud computing. In: Proceedings of the 2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT 2014), Astana, Kazakhstan, 15–17 October 2014, pp. 244–248 (2014)
24. Boranbayev, A., Boranbayev, S., Nurusheva, A.: Analyzing methods of recognition, classification and development of a software system. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) Intelligent Systems and Applications, vol. 869, pp. 690–702. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-01057-7_53
25. Boranbayev, A.S., Boranbayev, S.N.: Development and optimization of information systems for health insurance billing. In: 7th International Conference on Information Technology: New Generations, ITNG 2010, pp. 1282–1284 (2010)
26. Akhmetova, Z., Zhuzbayev, S., Boranbayev, S., Sarsenov, B.: Development of the system with component for the numerical calculation and visualization of non-stationary waves propagation in solids. *Front. Artif. Intell. Appl.* **293**, 353–359 (2016)
27. Boranbayev, S.N., Nurbekov, A.B.: Development of the methods and technologies for the information system designing and implementation. *J. Theor. Appl. Inf. Technol.* **82**(2), 212–220 (2015)
28. Boranbayev, A., Shuitenov, G., Boranbayev, S.: The method of data analysis from social networks using apache hadoop. In: Latifi, S. (ed.) Information Technology – New Generations. AISC, vol. 558, pp. 281–288. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-54978-1_39
29. Boranbayev, S., Nurkas, A., Tulebayev, Y., Tashtai, B.: Method of processing big data. In: Latifi, S. (ed.) Information Technology – New Generations. AISC, vol. 738, pp. 757–758. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-77028-4_99
30. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Estimation of the degree of reliability and safety of software systems. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) FICC 2020. AISC, vol. 1129, pp. 743–755. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39445-5_54
31. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Development of the technique for the identification, assessment and neutralization of risks in information systems. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) FICC 2020. AISC, vol. 1129, pp. 733–742. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-39445-5_53
32. Boranbayev, A., Boranbayev, S., Nurusheva, A., Seitkulov, Y., Nurbekov, A.: Multi criteria method for determining the failure resistance of information system components. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) FTC 2019. AISC, vol. 1070, pp. 324–337. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-32523-7_22

33. Boranbayev, A., Boranbayev, S., Nurbekov, A., Taberkhan, R.: The software system for solving the problem of recognition and classification. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) CompCom 2019. AISC, vol. 997, pp. 1063–1074. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-22871-2_76
34. Akhmetova, Z., Boranbayev, S., Zhuzbayev, S.: The visual representation of numerical solution for a non-stationary deformation in a solid body. In: Latifi, S. (ed.) Information Technolog: New Generations, pp. 473–482. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32467-8_42
35. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Evaluating and applying risk remission strategy approaches to prevent prospective failures in information systems. In: Latifi, S. (ed.) 17th International Conference on Information Technology–New Generations (ITNG 2020). AISC, vol. 1134, pp. 647–651. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43020-7_87
36. Boranbayev, A., Boranbayev, S., Nurbekov, A.: A proposed software developed for identifying and reducing risks at early stages of implementation to improve the dependability of information systems. In: Latifi, S. (ed.) 17th International Conference on Information Technology–New Generations (ITNG 2020). AISC, vol. 1134, pp. 163–168. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-43020-7_22
37. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Java based application development for facial identification using OpenCV library. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2020. AISC, vol. 1251, pp. 77–85. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55187-2_8
38. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Measures to ensure the reliability of the functioning of information systems in respect to state and critically important information systems. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2020. AISC, vol. 1252, pp. 139–152. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55190-2_11
39. Boranbayev, A., Boranbayev, S., Nurbekov, A.: Development of a hardware-software system for the assembled helicopter-type UAV prototype by applying optimal classification and pattern recognition methods. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) SAI 2020. AISC, vol. 1229, pp. 380–394. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52246-9_28
40. Boranbayev, A., Shuitenov, G., Boranbayev, S.: The method of analysis of data from social networks using rapidminer. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) SAI 2020. AISC, vol. 1229, pp. 667–673. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52246-9_49
41. Askar, B., Seilkhan, B., Yuri, Y., Yersultan, T.: Methods and software for simulation of optimal renewal of capital assets. J. Theor. Appl. Inf. Technol. (JATIT) **98**(21), 3545–3558 (2020)



Automated Data Processing of Bank Statements for Cash Balance Forecasting

Vlad-Marius Griguta¹(✉), Luciano Gerber², Helen Slater-Petty¹, Keeley Crocket²,
and John Fry³

¹ AccessPay, Manchester M1 4BT, UK

vlad-marius.griguta@accesspay.com

² Manchester Metropolitan University, Manchester M15 6BH, UK

³ Bradford University, Bradford B7 1DP, UK

Abstract. The forecasting of cash inflows and outflows across multiple business operations plays an important role in the financial health of medium and large enterprises. Historically, this function was assigned to specialized treasury departments who projected future cash flows within different business units by processing available information on the expected performance of each business unit (e.g. sales, expenditures). We present an alternative forecasting approach which uses historical cash balance data collected from standard bank statements to systematically predict the future cash positions across different bank accounts. Our main contribution is on addressing challenges in data extraction, curation, and pre-processing, from sources such as digital bank statements. In addition, we report on the initial experiments in using both conventional and machine learning approaches to forecast cash balances. We report forecasting results on both univariate and multivariate, equally-spaced cash balances pertaining to a small, representative subset of bank accounts.

Keywords: Time series forecasting · Cash flow forecasting · Data wrangling

1 Introduction

Cash flow forecasting is a critical task for corporations of all sizes and across the whole spectrum of business activities. The more diverse these activities are, the more demanding it is for the company stakeholders to make informed financial decisions. Teams of experienced treasurers are involved in estimating the future available cash of corporations and making investment decisions based upon these estimations. Every hour spent on investigating the financial information results in a non-negligible opportunity cost to the business stakeholders, especially in volatile times when the distribution of resources needs to be closely monitored. The benefits of using analytical methods to issue cash flow forecasts based on historical data are, therefore, twofold. Firstly, there is the potential to improve the forecasting accuracy, which optimizes the resource allocation across the business and, secondly, the potential to improve the performance of the treasury departments, shifting the focus from low-yield data collection tasks to lucrative investment decision making.

This paper addresses the problem of forecasting daily cash balances of corporate bank accounts by modeling the data as equally-spaced time series. In line with the typical cash operations of medium and large enterprises, the forecasting time horizon of our predictive models has been set to one full month (22 business days). This requirement for predictions over larger time spans seems to be a distinguishing feature and additional challenge addressed in this work, when compared to other common financial time series modelling tasks (e.g., stock prices movement). The data in this research is extracted from bank statements, which are collected through the SWIFT network (see Sect. 4.1). The information contained in bank statements is aimed at providing live cash visibility and transparency across bank accounts. These are a standardized form of communicating financial information, which means that a forecasting system that consumes bank statements can be used by a large number of companies. By focusing on reporting the overall liquidity within the account, bank statements often neglect reporting information at the transactional level, which impedes the flexibility of analytical forecasting methods. In our approach, we have identified and addressed some significant challenges (Sect. 4) in data collection and pre-processing, as well as with forecasting itself. Some of these challenges, such as inconsistency of transactional data reconciliation, irregularity of the statement issue time and missing data, are faced while reconstructing the cash balance data from the bank statements of different accounts. Other challenges revealed in the modeling process are related to the reduced historical data, operational outliers and pattern changes in cash balances. The main contribution of this paper is the assessment of these practical challenges and the proposal of solutions to alleviate them, with a view of scoping the prediction of cash balances based on bank statement data as a pure time series forecasting problem. By addressing the challenges, this paper exemplifies the use of both conventional (SARIMA, TES) and machine learning (ANN) models to issue cash balance forecasts in an automated and scalable manner. The scalability of the approach presented in this paper is compared to the historical (and still conventional) method used in cash flow forecasting. This method requires manual data aggregations and domain experts to estimate the expected performance of different business units within an organisation.

This paper is organized as follows: Relevant terminology is first introduced in Sect. 2, followed by a description of the data in Sect. 3. Section 4 describes data challenges associated with account balance forecasting. Section 5 presents related work on both conventional and machine learning methods applied to the problem of time series forecasting. Section 6 describes the experimental methodology used to compare the performance of several conventional and machine learning methods across cash balance accounts selected to illustrate the identified challenges. Section 7 presents the conclusions and future directions.

2 Terminology and Notation

A time series dataset is a series of values of one variable (univariate) or multiple variables (multivariate) that are organized in an ordered structure provided by the time component of the series. The time component of a time series not only enriches the series with information, but also sets constraints on the dependencies between the values of the

variables in the series. In that respect, all entries of a time series are interdependent, which constraints the sampling methods that can be applied to the data.

A forecast of a time series is defined as an ordered prediction of the future values of the series. We define a time series y_k , where y_k takes values from the ordered group y_1, y_2, \dots, y_n . A forecast $F_{n+1}, F_{n+2}, \dots, F_{n+m}$ of the series is defined as a prediction the values of the series y_k , over the period of $n, n + m$ days, where m is the forecasting horizon. The accuracy of a forecast is inferred from the deviation of the forecast from the actual values of the series over the forecasting horizon. There are multiple functions that can be used to compute the deviation. In this paper, we used the normalized root mean squared error (NRMSE) and the symmetric mean absolute percentage error (SMAPE) defined below:

$$NRMSE = \sqrt{\frac{\sum_{i=n+1}^{n+m} (F_i - y_i)^2}{m * \sum_1^n (y_i - \bar{y})^2}}, \quad (1)$$

$$SMAPE = \frac{100\%}{m} \sum_{i=n+1}^{n+m} \frac{|F_i - y_i|}{(|F_i| + |y_i|)/2}. \quad (2)$$

A transfer function f is used to map a subset of the past values of the series to the forecasted values. Depending on the forecasting methods used, the subset of past values can vary in size. A good way to choose the subset size is by analyzing the autocorrelation function of the series. The autocorrelation function is the correlation of the signal with a lagged copy of itself as a function of the lagging steps. A strong correlation for a certain number of lagged steps l indicates that the value of the k^{th} entry in the series has a strong influence on the $(k + l)^{\text{th}}$ entry, and therefore can be used to predict it. Similarly, a rapid drop in the autocorrelation function at lag l' indicates that the entries past the l'^{th} do not influence the prediction power of the transfer function, suggesting feeding the function a subset of l' series entries. So far, only the univariate time series forecasting problem has been discussed. A multivariate time series problem can contain, along the target series y_k , both endogenous and exogenous variables. An input variable is exogenous if it influences the target variable without being influenced by it; and endogenous if it can be influenced by the target variable. For example, the day in the month might influence the corporate cash balance due to the seasonality of cash operations, however, the cash balance does not influence what day it is. In contrast, the largest daily transaction within a bank account influences and is influenced by the daily cash balance of the account.

3 Data Description

The format of the data collected for this work is standardized by the Society for Worldwide Interbank Financial Telecommunication (SWIFT), the provider of a global network used for financial transactions. Depending on the scope of the information transferred, SWIFT uses different messaging types (MT). The types referring to cash management are under the format MT9xx. The two messaging types used in this work are MT940 and MT942. MT940 is the format used for end-of-day bank account statements whereas

MT942 is the format used for intraday reporting. An MT940 statement includes the list of transactions having cleared during a business day and a MT942 statement includes a subset of the transactions cleared from the previous statement onwards. Although the standard of the messaging types for cash management services is ensured by SWIFT, different banking entities have different data submission conventions maintained within their various legacy systems.

The bank statements issued for corporate users are similar in format to the retail bank statements. They contain a header detailing the account name, identification code, date-time of issue and the opening balance and closing balance. The bulk statement contains the list of transactions that sum up to the difference between the closing and the opening balance. Each transaction has an entry date, value date, amount, and several optional references: funds code, transaction type, identification code, reference to account owner and information for account owner. A subset of relevant features synthesized from a statement is: “‘`datetime`’: ‘2020-01-30 21:30’, ‘`open`’: £400000, ‘`close`’: £387000, ‘`transactions`’: {‘`value date`’: ‘2020-01-31 00:00’, ‘`value`’: +£4000, ‘`identification code`’: ‘CHK’, ‘`ref`’: ‘12354538 Cheque Company A’, etc.}”.

This section outlined the richness of the information contained in a standard bank statement. Section 4 will discuss the specific challenges associated with the extraction of consistent and ordered cash balances from various bank statements.

4 Account Balance Data Challenges

The challenges presented by time series forecasting are well known. Fine-tuned models developed through conventional or machine learning methods suffer from time instability due to the lack of stationarity, degrees of uncertainty in historical time periods and the restrictions on train, test and validation splits due to time sequences, see e.g. [1]. Time Series data corresponding to financial transactions, however, pose additional difficulties, which do not seem to have been addressed in the academic literature yet.

4.1 Reconstruction Challenges

Inconsistency of Transactional Data Reconciliation. Some challenges of integrating merging financial time series data into a consistent format were discussed in [2]. Due to the global nature of many corporations, international payments and markets have an effect on statement data. Statements received from the bank are given relative timestamps based on time zones, meaning that collating transactions from multiple regions can lead to discrepancies. One such discrepancy often occurs when aggregating intraday statements (MT942) and reconciling against end of day statements (MT940), due to transactions having value dates past the issue date of the end of day statements.

Irregularity in Statement Time of Issue. In addition to reconciling transactional information within bank statements at the global market scale, a forecasting system needs to consider the temporal component of the cash balance time series obtained from bank statements. The temporal component of a statement of a bank account can be

inferred as the time of issue of the statement by the banking institution managing the account. Because different banking institutions have different legacy systems for collating transactional information into statements, the statement reporting offering varies across regions and legislations but also across banks within the same country. Most affected by this inconsistency are the intraday statements which need to be assigned an accurate date and time to allow for the reconstruction of the corresponding time series. With the statements arriving at different times during the day, only an intermittent time series can be reconstructed. Additionally, the average number of statements collected per day is between 2 and 5, limiting the potential of resampling methods as a solution to the series intermittence. Given the circumstances, we chose to de-scope the use of intraday statements from the current study.

Missing Data. Notwithstanding the irregularity in time of issue, assuming that there are a small number of transactions at the time when the end of day statement is collated (usually mid-night), the reconstructed time series of the end of day balances should be continuous and equally spaced. However, there are other factors that influence the data collection process, within a live environment. For example, there can be a temporal downtime of a service yielding interruptions or duplications of data that needs to be investigated manually. A specific example is the closing available balance which might be reported only sparsely or not reported at all for some bank accounts.

It is important to note that in other contexts that data seemingly missing at random can have a hidden serious bias (see e.g. [3]). Here, much of the data is missing purely due to different financial reporting conventions associated with the different accounts. Rather than being a specific problem with missing data per se, it is possible that there may be some hidden structure in the data associated with an intra-monthly effect identified by practitioners. This is something we wish to explore in a continuous-time model in future research [4].

4.2 Forecasting Challenges

Reduced Historical Data - Many of the state-of-art linear forecasting algorithms of univariate time series are based upon the extraction of seasonal patterns. These patterns are either extracted directly, in the case of stationary seasonality, or are indirectly trained upon through feeding the algorithms additional exogenous features of the time component (e.g. day of month, month of year, etc.). However, when the time series has less than one full period of a season, little information can be inferred on the seasonal pattern of the series. In the context of this study, many of the corporate accounts considered were recorded for between 6 to 9 months, which posed a challenge in modelling any potential pattern manifested for longer than a quarter. Additionally, depending on the commercial agreement between the client and their banking partners, the cash balances of the client's bank accounts are reported with different granularities. Some accounts are reported on every calendar day, accounts are only reported on working days, and others are reported at different days during the month. For example, there are accounts which are set to report only on the first Wednesday, Thursday and Friday of each month (Fig. 1). The resulting time series of cash balances is sparse, with the balance information

being populated at irregular time intervals that depend on the day of week rather than the calendar day. In both these situations, we chose to put the accounts with insufficient data points out of scope of this paper. Consequently, we excluded the bank accounts for which the reconstructed cash balance contained less than 15 data points per month (missing data) or less than 6 months' worth of data.

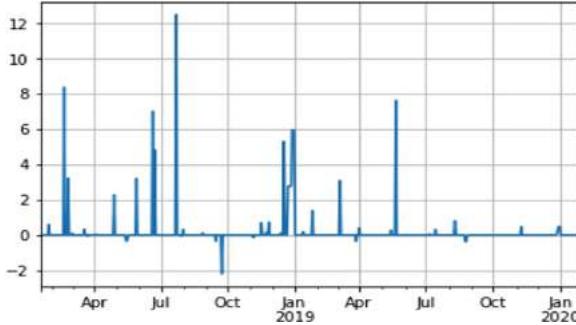


Fig. 1. Sparsely reported account. The y axis is a scaled true cash flow of the account.

Operational Outliers - Barring the stochastic component of the cash inflows and outflows, which are guided by external factors such as market volatility or client performance, the data contained within the statements issued for a business bank account is, a genuine representation of the cyclical business activities managed by the account. However, within the treasury management sector, it is often the case that seemingly random interventions of the domain specialists blurs away the valuable insights that the data has to offer (e.g. Fig. 2). Operations such as inter-subsidiary lending, long term investments, mergers or acquisitions are traced in the cash balance as irregular sparks or dips which are challenging to be picked up through univariate modelling of the time series. The immediate solution to the challenges posed by the operational outliers is to flag out and eliminate the transactions corresponding to these outliers from the reconstructed cash balance and only model those transactions that are inherent to the business operations performed through the account. However, due to the subjective nature of the treasury management decisions, flagging these manual interventions has proven to be challenging even for domain experts that are external to the company department making the treasury decisions. Applying tests based on statistical metrics such as the standard deviation did not yield an improvement in forecasting performance. In the future we plan to use more sophisticated metrics for identifying the operational outliers, including smoothing functions and designated anomaly detection models.

Pattern Changes in Cash Balances - Large enterprise clients managing multiple production lines, brands and businesses present an additional layer of complexity into their cash balance sheets. Due to the volatility of the different markets their products address, decisions of cash allocations are being made at an increased pace, resulting in bank accounts being opened, closed or put out of use at various times during the year. One example of a cash balance with pattern change is shown in Fig. 3. In this instance, the average cash balance suddenly drops during the period April to October

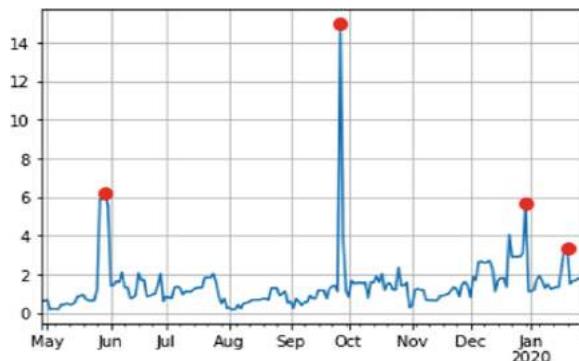


Fig. 2. Operational outliers revealed in an account sample. The y axis is a scaled true cash balance of the account.

2019 with no respective behavior for the same period of 2018. Similar anomalies in the cash balances were considered separately by applying transformations to the time series and optimizing the model hyper-parameters in a manual manner.

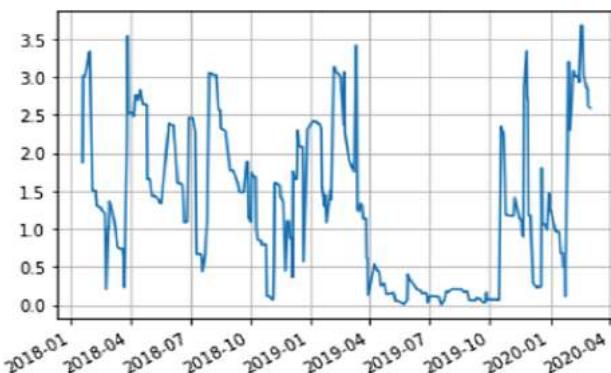


Fig. 3. Account sample demonstrating a pattern change in the period April - October 2019. The y-axis represents the scaled true cash balance of the account.

5 Data Description

This section provides an overview of related work using both conventional and machine learning methods for financial forecasting. Traditional methods include Naïve Forecasting, which uses the actual cash flow data from a previous time period as the forecast for the upcoming period (REF). Simple Moving Average Forecasting (SMAF) adds the recent closing prices, then divides the total by the number of time periods to calculate the average. Exponential Smoothing uses exponential functions to assign exponentially decreasing weights over time periods and is useful when the more recent past is likely to

have more impact on predicting a forecast than more historical past. For large historical cash balance datasets, autoregressive moving average (ARMA) and autoregressive integrated moving averages models (ARIMA) can be used to identify autocorrelations and are more suitable for long term forecasting. However, due to the challenges identified in Sect. 4, there is no one size fits all solution.

5.1 Conventional Approaches to Financial Forecasting

It is important to recognize that finance presents idiosyncratic forecasting challenges that are significant in their own right. The inherently stochastic nature of the subject is further exacerbated by additional sources of short-term randomness that are typically modelled using an unobserved stochastic volatility component. Conventional forecasting approaches such as ARIMA are also typically at odds with notions of market efficiency [5], theoretical options-pricing models constructed via foundational arguments such as absence of arbitrage [6] and the stylized empirical facts of financial time series [7]. Financial time series modelling is also an inherently specialist area. Commonly used ARCH/GARCH models for (financial) time series are equivalent to ARMA models for an unobserved volatility component. However, a range of different model variations are possible [8]. This sheer range of available models underscores the specialist nature of the subject. Further extensions of these classical models have been used in applications that variously account for further autocorrelations in the observed series [9] and for additional regime-switching effects [10]. However, within this, the need to account for unobserved volatility fluctuations remains paramount.

Other non-time-series approaches to forecasting, e.g. those based around the technical analysis methods popularized by practitioners (see e.g. [11]), are possible. The academic literature on technical analysis is voluminous, see e.g. [12] or [13] for a review. However, such approaches have yet to permeate mainstream finance. An exception is [14] who use localized regression approaches to gauge the plausibility of technical analysis strategies. Thus, this serves to motivate the study of machine-learning techniques within financial forecasting. As an illustration, [15] reviewed corporate cash flow forecasting using account receivable data collected through a specialized accounting software, which provides a richer view of the individual transactions.

5.2 Machine Learning Approaches to Financial Forecasting

The use of deep learning for time series prediction, in specific domains is not new, but remains challenging due to the need for extremely large datasets of high quality data and the lack of transparency in how decisions were made. One of the most targeted areas is stock market forecasting predicting stock prices in different time slice windows. The author in [16] created a deep learning framework which combined wavelet transforms, stacked auto-encoders and long-short term memory (LSTM) networks to predict six stock indices, one-step-ahead of the closing price. The author in [17] utilized LSTM networks for predicting out-of-sample directional movements for a number of financial stocks and outperform other methods such as random forests. The author in [18] proposed day-ahead multi-step load forecasting using both recurrent neural networks (RNN) and convolutional neural networks (CNN). In their work the use of the CNN model improved

the forecasting accuracy by 22.6% compared to the application of seasonal ARIMAX. However, the dataset used was concerned with predicting accurate building-level energy load forecasts which looked at how similarities within data space can be identified in financial forecasting.

Whilst Deep Learning has been successfully applied in many domains, it is not always successful. Small datasets do not tend to perform well, with research indicating that to be successful, millions of data points are required. Data quality is always an issue when applying machine learning, the generalization error of artificial neural networks can be improved by the addition of noise in the training phase. Consequently, this provides a barrier to be overcome with respect to forecasting financial time series. This may help to explain some of the data challenges described in Sect. 3. Despite some recent progress, explaining and interpreting models remains challenging. Ensuring the financial interpretability of the deep learning models constructed is thus far from being a foregone conclusion.

Ensemble machine learning models have also been used for financial forecasting in e.g. [19] and [20]. The author in [19] combined two traditional ensemble machine learning algorithms: random subspace and MultiBoosting to create a method known as RSMultiBoosting to try and improve the accuracy of forecasting the credit risk of small-to-medium companies. RSMultiBoosting outperformed traditional machine learning algorithms on small datasets and the ability to rank features according to the decision tree relative importance score improved accuracy. The author in [21] conducted a study investigating several models including deep and recurrent neural networks and the CART regression forest to examine non-linear relationships between input and output features on abnormal stock returns generated from earnings announcements based on financial statement data. The results indicated that non-linear methods could predict the direction of the “absolute magnitude of the market reaction to earnings announcements correctly in 53% to 59% of the cases on average.” The author in [21] with random forest approaches providing the best results. Whilst this is a reasonable result, it highlights the issues of data quality (as discussed in Sect. 2) and its impact on whether an account is forecastable.

6 Experimental Comparison

This section provides a comparative analysis of the performance of different time series forecasting techniques on a subset of anonymized time series that replicate real bank account cash balances. The sampling and pre-processing procedures are explained in subsection A. We report a novel approach of enriching the univariate time series pertaining to cash balance data by aggregating the transactions pertaining to bank statements by various statistical metrics (e.g. standard deviation). Subsection B describes the forecasting methodologies, beginning with the univariate approaches (SARIMA and TES), then enriching SARIMA with multivariate exogeneous constraints, and ultimately leveraging the multivariate input via a neural network architecture.

6.1 Dataset Description

To share the learnings gained from forecasting cash balance in various bank accounts, a sample time series dataset representative of cash balance data was created. The dataset

consists of collections of time series corresponding to daily cash balances and some additional exogenous and endogenous variables. As described throughout the section on the data challenges, there are multiple granularities in which the account balance statements are recorded. The examples selected for this work are those for which the reconstructed time series contains at least one data point per business day. An additional level of complexity is given by the possibility of some accounts to consist of a group of individual bank accounts.

A sparse time series is created by collating the closing balance amounts with corresponding value dates. To provide a consistent view over the balances of multiple accounts of a client, the closing balances are converted to a currency of choice. The missing values in the sparse time series are then forward filled to the granularity set for the account (per business day or daily). The reason for filling the values with previous valid entries (forward filling) is that it is presumed that in the valid dates when the balance is not reporting, there were no cash movements. In the case of groups of accounts, the date range is initially established as the minimum and maximum dates reported by any of the bank accounts in the group. The missing values in each individual bank account are then filled, initially forward and then backward as well, to cover the period between the earliest statements of the group and the individual earlies statement. Subsequently, the individual continuous time series are summed up to the grouped time series. The exogeneous variables obtained from the time component of each time series are then computed. Endogenous variables obtained through applying various statistical aggregations to the transactional statement data are also used to predict the cash balance. However, due to potential clashes with the IP of AccessPay, the aggregation methods will not be discussed explicitly. The sample dataset used throughout the paper represents a selection of 6 bank account aggregates with end of day cash balances reported in each business day during the period 18 June 2019 – 27 January 2020. Figure 4 below shows each time series collected. A train-test split was applied, where the size of the test set is equal to the forecasting horizon of one month.

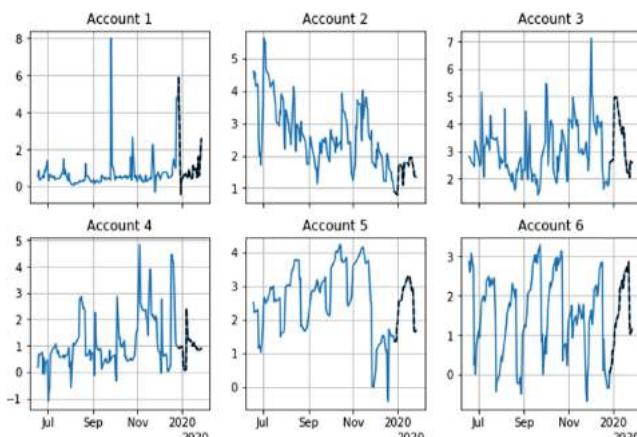


Fig. 4. Cash balance series of the selected accounts. The train set is shown in blue and the test set is shown in black. The y axis represents the scaled true cash balance of each account.

6.2 Experimental Results

The results of the forecasting experiments are separated in three sections. The first section discusses the problem of univariate time series forecasting via conventional methods, ARIMA and TES. In the second section we propose an extension of the ARIMA to account for multivariate input. The third section discusses the results of a machine learning approach based upon ANNs.

Conventional Univariate Time Series Forecasting. The conventional time series forecasting algorithms referred throughout this paper are the Seasonal Autoregressive Integrated Moving Average (SARIMA) and the Triple Exponential Smoothing (TES). The SARIMA model has seven hyper-parameters, one for the seasonal differentiation term, three for modelling the seasonal component of the series and three for modelling the remainder of the series. In the experiments undertaken for this paper, these parameters were determined through experimentation against the AIC score, leading to the SARIMA (1,0,1)(1,1,1)22. The value of 22 for the seasonal differentiation was chosen as the main number of business days in a month.

The Triple Exponential Smoothing features four hyper-parameters. These are the trend type, the damping of the trend, the seasonal type and the seasonal differentiation term. Based upon heuristics, it was found that the best set of values for these parameters are: trend: additive, damped: False, seasonal: additive and seasonal differentiation: 22.

The first three rows in Table 1 detail the performance of the two methods over the account samples, as compared to the Naive Average benchmark. While there are some accounts for which the TES prevails both the benchmark and the SARIMA model, overall, only SARIMA overcomes the benchmark in a consistent manner.

Table 1. Aggregate accuracy metrics on the account samples

Account	A1	A2	A3	A4	A5	A6	Mean
Root mean squared error							
Naïve Mean	1.27	4.47	1.12	0.47	0.66	0.85	1.47
SARIMA	0.67	0.73	0.93	4.25	0.70	0.42	1.28
TES	5.51	0.3	8.66	2.22	0.17	1.05	2.99
MULTI SARIMA	0.67	0.56	0.62	4.67	1.55	0.53	1.43
ANN	1.42	3.12	0.44	1.38	1.33	0.43	1.35
Symmetric mean absolute percentage error							
Naïve Mean	53.17	60.38	26.86	33.98	22.84	50.87	41.35
SARIMA	64.44	26.68	24.37	61.79	29.59	51.91	43.13
TES	129.2	24.37	132.0	73.78	10.82	81.01	75.24
MULTI SARIMA	63.64	27.73	22.68	60.51	47.26	55.19	46.17
ANN	59.02	52.71	16.65	74.60	31.22	40.93	45.87

To understand the gains and failures of the conventional univariate time series forecasting methods, we looked at the extreme cases for which the methods either outperformed or underperformed the benchmark by a considerable margin. From Table 1, these are account 2 (Fig. 5) and account 3 (Fig. 6). On the second account sample, both ARIMA and TES yielded accuracies exceeding the benchmark as measured by both the NRMSE and the SMAPE metrics. Noticeably, the NRMSE score of TES was the global minimum across all methods and accounts tested.

A different outcome was observed for Account 3. In this example, the vague monthly seasonality is only captured by the ARIMA method while TES seems to be tricked by the outlier around December into predicting a decreasing trend across January. To conclude, the univariate models are unstable and fail to generalize on the multitude of cash balance series. While some level of progress is achieved for a subset of accounts through these methods, they would ultimately be overwhelmed by the unresolved challenges discussed in Sect. 2, in particular the Operational Outliers. As these outliers appear into the cash balances as a result of the institutional decisions that are made based upon transactional data, it is hoped that through making use of the information contained within the transactional data more insights could be drawn to support the unidimensional forecasting.

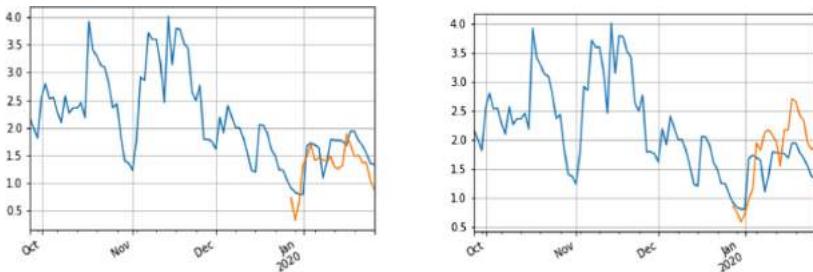


Fig. 5. Account 2 – Conventional forecasting outperformed the benchmark. The y Axis represents the scaled true cash balance of the account.

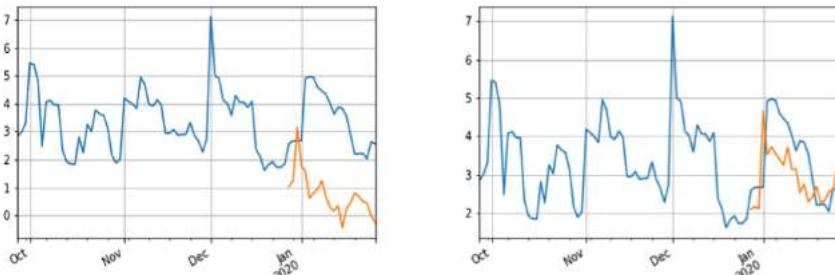


Fig. 6. Account 3 – Only SARIMA outperformed the benchmark. The y axis represents the scaled true cash balance of the account.

Conventional Multivariate Time Series Forecasting. As detailed in the dataset description (Sect. 6.1), a multivariate dataset was obtained through extracting information from transactional data contained within the bank statements. A series of aggregation techniques were applied on the transactional data, each based upon the different types of transactions and correlations between them. The aggregates obtained in a form of sparse time series are forward-filled to ensure there is no effect of future values on the past entries. As the sum of the transactions within a day reconcile the end of day cash balance, the aggregate transactional time series represent a set of endogenous variables to the target variable. Therefore, these series cannot be directly used as exogenous constraints to the SARIMA model. The solution implemented in this paper was to shift the time component of the transactional aggregates forward by the size of the test set. In other words, for example, the aggregates computed for the July cash balances were used as exogenous constraints for predicting the cash balance during August (Fig. 7).

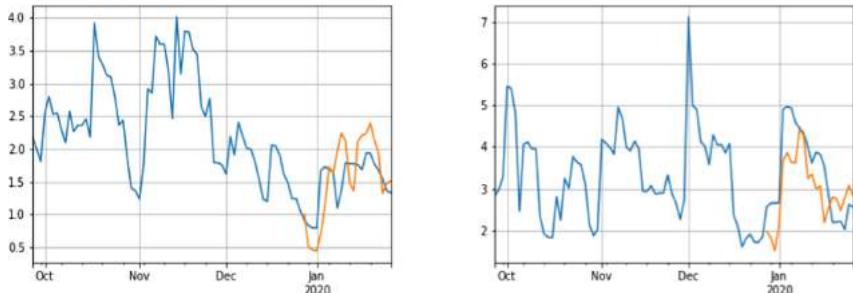


Fig. 7. Performance of SARIMA with transactional exogenous variables on accounts 2 and 3. The y axis represents the scaled true cash balance of each account.

Machine Learning Approaches. The improvement in accuracy through the use of exogenous constraints based upon past transactional data indicates that the transactional data could be leveraged to train endogenous regressors on top of the univariate signal from the cash balance. A family of models that were shown to be able to map multivariate inputs to time ordered outputs are the neural networks (e.g. [22]). The assumption that the transactional aggregates retain information about future cash balances, a stacked ensemble of a dimensionality reduction algorithm and an artificial network architecture was built. Due to the commercial nature of the experiment reported in this paper, the exact architecture of the neural network (ANN) cannot be revealed.

We report that the ANN architecture outperforms SARIMA with transactional exogenous constraints judged by both metrics used in this experiment. Compared against the univariate SARIMA, the machine learning method still underperforms, albeit by a small margin of 0.07 in the normalized root mean squared error metric. By comparing the individual performance over each account sample we can get a clearer picture of the difference between conventional and machine learning models.

Figure 8 presents a comparison of the accuracy of the two best performing models, univariate ARIMA and multivariate ANN, over the same two account samples discussed

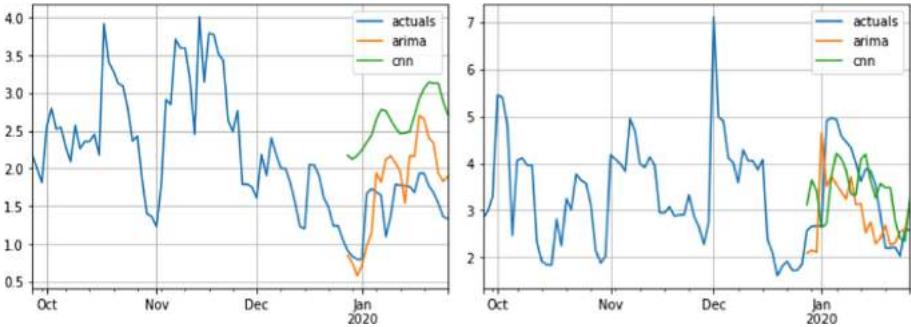


Fig. 8. Performance comparison of the univariate ARIMA and the multivariate NN methods on Account 2 (0.73 vs 3.11) and Account 3 (0.93 vs 0.44). The y axis represents the scaled true cash balance of each account.

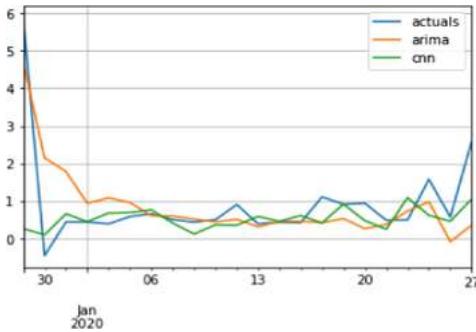


Fig. 9. Performance comparison of univariate ARIMA and multivariate NN over Account 1 (0.67 vs 1.42). The y axis represents the scaled true cash balance of the account.

in previous sections. The 3 month historical actuals of the cash balances are included to give a view of the most recent trend in the series. Noticeably, while the ANN method outperforms by a large margin ARIMA on the Account 3, the performance over the second account is poor. We attributed the underperformance to the changing trend of the second account, which the ANN could not capture. In Fig. 9 we compare the two models over a different bank account which contains a salient operational outlier. Similar to the case of the second account, the neural networks could not learn the rapidly changing pattern of Account 1. However, when eliminating the very first datapoint in the series, the root mean squared error drops from 1.42 to 0.45, whereas the SARIMA model yields the same error of 0.66 (Table 1). These observations suggest that, given the rapidly changing trends within the series, the ANN architecture can learn the smoothly varying features better than the conventional methods.

7 Conclusions and Further Work

In this paper we presented an overview of the challenges of understanding, collating and exploring account balance data pertaining to private enterprises with a view of forecasting

their future cash balances. We showed several techniques for addressing these challenges, which allowed us exemplify the use of both conventional and machine learning techniques for cash balance forecasting. Additionally, we performed a comparative analysis of conventional and machine learning based time series forecasting models on a representative subset of bank accounts. The intermediate results portrayed a fair competition between Seasonal Auto Regressive Moving Average and Neural Network Architectures, indicative of the stochastic nature of enterprise account cash balances. Permanent collaboration with field experts, either internal or external (banks and customers), and with the architects of the legacy code used for parsing the SWIFT statements is the ultimate solution to alleviating a number of challenges discussed in this paper. In the future, we plan on exploring the transactional features in more depth to get an understanding of the way they infer the future values of the end of day cash balance. Additionally, through an improved collaboration with the domain specialists, we aim at limiting the influence of the challenges emphasized in this paper.

Acknowledgments. The authors would like to express their gratitude to the two anonymous referees for the helpful and supportive comments received.

References

1. Giles, C.L., Lawrence, S., Tsoi, A.C.: Noisy time series prediction using recurrent neural networks and grammatical inference. *Mach. Learn.* **44**, 161–183 (2001)
2. Katselas, D., Sidhu, B., Yu, C.: Merging time-series Australian data across databases: challenges and solution. *Account. Finan.* **56**, 1071–1095 (2016)
3. Shang, Y.: Subgraph robustness of complex networks under attacks. *IEEE Trans. Syst. Man Cybern. Syst.* **49**, 821–832 (2019)
4. Fry, J., Griguta, V.-M., Gerber, L., Slater-Petty, H., Crockett, K.: Stochastic modelling of corporate accounts. Preprint (2021)
5. Fama, E.: Efficient capital markets: a review of theory and empirical work. *J. Finan.* **25**, 383–417 (1970)
6. Merton, R.C.: The theory of rational options pricing. *Bell J. Econ. Manag. Sci.* **4**, 141–183 (1973)
7. Cont, R.: Empirical properties of asset returns: stylized facts and statistical issues. *Quant. Finan.* **1**, 223–236 (2001)
8. Hentschel, L.: All in the family: nesting symmetric and asymmetric GARCH models. *J. Finan. Econ.* **39**, 71–104 (1995)
9. Katsiampa, P.: Volatility estimation for Bitcoin: a comparison of GARCH models. *Econ. Lett.* **158**, 3–6 (2017)
10. Walid, C., Chaker, A., Masood, O., Fry, J.: Stock market volatility and exchange rates in emerging countries: a Markov-state switching approach. *Emerg. Mark. Rev.* **12**, 272–292 (2011)
11. Meyers, T.A.: The Technical Analysis Course, 4th edn. McHraw-Hill, New York (2011)
12. Park, C.-H., Irwin, S.H.: What do we know about the profitability of technical analysis? *J. Econ. Surv.* **21**(4), 786–826 (2007). <https://doi.org/10.1111/j.1467-6419.2007.00519.x>
13. Nazário, R.T.F., e Lima, J.L., Sobreiro, V.A., Kimura, H.: A literature review of technical analysis on stock markets. *Q. Rev. Econ. Finan.* **66**, 115–126 (2017). <https://doi.org/10.1016/j.qref.2017.01.014>

14. Lo, A.W., Mamaysky, H., Wang, J.: Foundations of technical analysis: computational algorithms, statistical inference and empirical investigation. *J. Finan.* **55**, 1705–1765 (2000)
15. Weytjens, H., Lohmann, E., Kleinstuber, M.: Cash flow prediction: MLP and LSTM compared to ARIMA and Prophet. *Electron. Commer. Res.* (2019)
16. Bao, W., Yue, J., Rao, Y.: A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLOS ONE* **12**(7), e0180944 (2017). <https://doi.org/10.1371/journal.pone.0180944>
17. Fischer, T., Krauss, C.: Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **270**(2), 654–669 (2018). <https://doi.org/10.1016/j.ejor.2017.11.054>
18. Cai, M., Pipattanasomporn, M., Rahman, S.: Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques. *Appl. Energy* **236**, 1078–1088 (2019)
19. Zhu, Y., Zhou, L., Xie, C., Wang, G.J., Nguyen, T.V.: Forecasting SMEs' credit risk in supply chain finance with an enhanced hybrid ensemble machine learning approach. *Int. J. Prod. Econ.* **211**, 22–33 (2019)
20. Salas-Molina, F.: Fitting random cash management models to data. *Comput. Oper. Res.* **106**, 298–306 (2019)
21. Amel-Zadeh, A., Calliess, J.-P., Kaiser, D., Roberts, S.: Machine Learning-Based Financial Statement Analysis, 15 January 2020
22. Akram, M., El, C.: Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. *Int. J. Comput. Appl.* **143**(11), 7–11 (2016)



Top of the Pops: A Novel Whitelist Generation Scheme for Data Exfiltration Detection

Michael Cheng Yi Cho^(✉), Yuan-Hsiang Su, Hsiu-Chuan Huang,
and Yu-Lung Tsai

Information and Communication Security Technical Laboratories,
Telecommunication Laboratories, Chunghwa Telecom, Taoyuan, Taiwan
`{michcho,yhsu,pattyh,tyl}@cht.com.tw`

Abstract. As intellectual property and private data are stored in digital format, organizations and companies must protect data from possible digital data exfiltration. Prevention and detection are common approaches to combat against data exfiltration. Regardless of prevention measures, detection approach must be implemented to strengthen data protection. Analyzing network data to find abnormal behavior, namely network traffic anomaly detection, is an effective detection approach. One of the challenges in network traffic anomaly detection is benign data or data noise filtering. A good data filter will improve computation performance and detection accuracy. In this paper, we proposed a whitelist generation scheme to generate data filter for data exfiltration detection. Our scheme leverages kernel density estimator with numerous computed network feature indexes to generate a popularity-based data filter. We use real corporate network data to evaluate data filter efficiency. After evaluation, our scheme generates a whitelist that filters out twice as more data in comparison with the conventional whitelist generation method. Furthermore, detection accuracy is also improved when evaluating against two data exfiltration detection algorithms.

Keywords: Whitelist · Data filter · Data exfiltration · Data leakage

1 Introduction

Data exfiltration incidents are much more organized nowadays giving its economic value. Stealing intellectual properties, personal identifies and data, etc. are often the motivation of organized cyber crimes. Even top financial companies like Equifax and Deloitte have fallen victims of such incidents [1]. Giving that most of information has gone digital, companies must not overlook the importance of digital data protection.

There are prevention and detection solutions to mitigate data exfiltration problem. Solutions like data access control, and data encryption are popular prevention scheme. These approaches only control access privilege to the protected

data. It cannot detect data misuse [2]. Detection mechanism, on the other hand, is adopted to strengthen data protection against data exfiltration. Anomaly detection is an active research domain due to recent raise of machine learning research. Giving network is one of the main medium for data exfiltration, network-based anomaly detection is one of active research domain. Network-based anomaly detection [3–5] leverages network features and machine learning to detection possible malicious activities which may cause data exfiltration.

Data filtering is a common data preprocessing approach in network-based anomaly detection. It is used to filter out benign data before detection model training process and/or during detection process. A good data filter will enhance computational performance and improve detection accuracy. Whitelist is a common method to construct data filter. Two common approaches are used to produce whitelist, namely domain knowledge heuristics and popularity heuristics. Domain knowledge heuristics [6, 7] requires expert knowledge of malicious behavior and training data. Therefore, it is hard to replicate whitelist generation procedure from research to research.

Popularity heuristics is another feasible solution since anomaly detection is based on rare event detection. This method generates whitelist by summarize common events (popular events) [4, 5, 8]. It gathers resources from web traffic statistics website (e.g. Alexa [9, 10]) or obtain the top-N most visited website from training data statistics to construct whitelist. Resource from the web suffers locality problem giving whitelist is not customized to the training data. The second approach often neglect to evaluate the effectiveness of top-N configuration (i.e. the approach to find a good N value), since whitelist generation is not the main objective of the research.

Popularity heuristics approach is a good approach, but it is not an easy problem to summarize a middle-to-large network heuristically. Part of our corporate network consist of over 10 thousand of clients that generate 40 million of connection in daily basis. It is difficult to develop a solution based on heuristics approach. If we could derive a systematic solution that overcomes locality and top-N configuration problem, anomaly detection research can benefit from such solution. For that, we propose a scheme that is based on popularity concept, and it uses statistic method to select popular domains to serve as whitelist. We name the scheme as TOP (top of the pops) since it adopts the concept of selecting most popular domains from the given network data.

In contrast to popularity heuristics, TOP systematically takes raw network data as input, compute feature indexes, using statistics to select whitelist candidate base, and expand from whitelist candidate base to output a reliable whitelist for data exfiltration detection algorithms. According to evaluation, TOP filtered out more data (twice as more) in comparison with the conventional popularity heuristics. As a results, TOP assists data exfiltration detection algorithms to perform better in detection accuracy and computation speed.

The remainder of this paper is organized as follows: Sect. 2 reviews related research regarding to whitelist generation and usage. Section 3 explains scheme structure and the algorithms used in the proposed scheme. Section 4 evaluates

proposed scheme against conventional popularity heuristics using two different data exfiltration algorithms with real network traffic dataset. Lastly, Sect. 5 concludes this research work.

2 Related Work

As for related works, we have surveyed whitelist generation approaches in different malicious network behavior detection domains and categorized the approaches based on adopted strategies. The strategy bases are domain knowledge-based, popularity-based, and statistical-based.

For domain knowledge-based approach, Hwang et al. [6] proposed a 3-tier IDS (Intrusion Detection System) using data mining. One of the tier uses whitelist filtering to filter out benign network traffic. A filter is built based on feature profiling. If observed network traffic does not fall into the profile, it is considered as abnormal traffic. This approach may trigger false negative if adversaries leverage popular web service to carry out malicious activity. Strayer et al. [7] introduced a botnet detection scheme that uses data filtering technique to filter out data noise. The technique is based on domain knowledge of malicious characteristic. It is a heuristic data filtering approach based on detection application. Take-mori et al. [11] proposed whitelisting approach for bot detection. A whitelist is constructed per work station/server unit. An observation period is required to construct the whitelist. This whitelist construction technique is difficult to deploy in large scale network. Furthermore, it is rather difficult to make sure no malware infection occurs during the observation period.

As for popularity-based approach, research leverages well known popular web service list gathered from experience and Internet resource. Yan et al. [4] constructed customized whitelist based on popularity statistics of network traffic, referer header, raw IP addresses. They also applied domain-folding to expand the coverage of whitelist. Gu et al. [8] proposed a bot C&C (command and communication) channel detection system based on network statistics correlation. The system also uses two types of whitelist, namely, hard and soft whitelist, to filter out data noise. A hard whitelist is based on popular web services (e.g. Google, Yahoo, etc.). A soft whitelist is obtained after data is analyzed as benign network traffic. Since anomaly detection detects rare behavior, it makes sense to filter out common behavior (visits to popular web services). However, popularity definition discussion is out the scope of these research.

Lastly, statistical-based approach summarize network features to distinguish normal and abnormal behavior. Byakko [12] is a whitelist generation approach based on data distribution. Byakko leverages deviation of expected distribution to determine data noise. A host is considered noise if feature distribution of the training data and testing data is similar. Byakko only works on a set features of a host that has stable distribution. Therefore, Byakko deployment will need experts with domain knowledge to preprocess the study data.

Although whitelist is not the main objective in most of the introduced research, it still serve an important role in data preprocessing. A good whitelist

will increase detection accuracy and reduce the computation time. However, most of the introduced research rely on expert's domain knowledge or popularity heuristic to generate a whitelist. Our goal is to propose a systematic whitelist generation scheme with minimum domain knowledge of deployed security application. The resulting generated whitelist is close to the study target which will yield better result in data filtering.

3 Proposed Scheme

Before proposed scheme is presented, we will begin with web sites in interest for our problem domain. In malicious network traffic analysis domain, whitelist candidates are often referenced as a popular web site. However, some popular web site may not be a good whitelist candidate depending on security application. In our case, online storage web service is not a good whitelist candidate for data exfiltration detection application. Therefore, we need to define whitelist candidate targets, so we can evaluate the generated whitelist candidates. The following items will be revisited in the evaluation subsection:

- Inclusive Domains
 - Popular domains with integrated services such as default page, search engine, news, portals, etc.
 - Domains that are work related, such as corporate sites.
 - Domains used for software update purpose.
 - Advertisement and client behavior tracking background network traffic when visiting a domain.
- Exclusive Domains
 - Well-known online storage domains.
 - Any domains that have extensive upload traffic.

Figure 1 depicts the workflow of TOP. Raw network proxy log in passed into three major components, namely, feature computation, whitelist candidate computation, and whitelist candidate expansion, to generate a list of whitelist candidates. Feature computation is responsible for feature extraction and feature indexes computation. As a result, two sets of feature indexes are passed into whitelist candidate computation component to generate two sets of whitelist candidates. Finally, whitelist candidate expansion component is in charge of expanding whitelist candidates to expand data filtering coverage. Each of the major components is explained in greater detail of the following subsections.

3.1 Feature Computation

The feature component consists two functionalities, namely, feature extraction and feature indexes computation, and outputs two set of feature indexes to serve as input for whitelist candidate computation (see Fig. 1). Feature extraction component extract client identification, visited domain, upload bytes, and

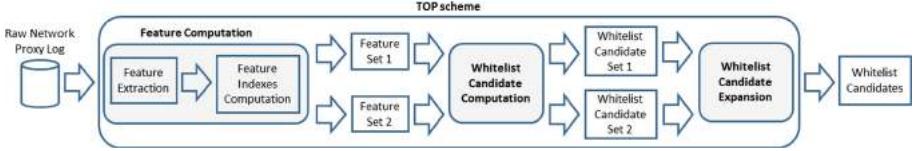


Fig. 1. Proposed scheme workflow.

download bytes. Feature indexes computation will use these features to compute two sets of feature indexes to serve as input of whitelist candidate computation.

The first set of feature indexes is number of accessed clients, number of network traffic flows, total downloaded bytes, and total uploaded bytes per domain in daily basis. To prevent rare event occurrence in the training data, we only select domains that are visited in all of the training data (training data are separated by day) for feature indexes computation. The feature indexes are used as indicator of a popular domain in the training data.

The second set of feature indexes uses access term frequency and upload-download ratio to rate a domain in daily basis. Feature index computation is as following:

$$TF_{ij} = \frac{1}{\text{count}(J_i)} \quad \text{where } \Rightarrow i \in I, \Rightarrow j \in J \quad (1)$$

Let I be clients and J be visited domains from the sampling data. Equation 1 computes visited domain TF (Term Frequency) for each client and domain pair. Whenever a distinctive domain is visited by a client, the corresponding TF_{ij} is a fraction of total distinctive domain visited by that client. TF_{ij} receives a higher value if client i only has a few distinctive network connections. A small set of distinctive domain connections means that the client is less likely to be operated by human, and the generated network traffic could be background noises generated by software updates. Therefore, such domains will receive a higher score.

$$UDR_j = \sum_{i=0}^I \frac{DL_{ij}}{UL_{ij}} \quad \text{where } \Rightarrow i \in I, \Rightarrow j \in J \quad (2)$$

$$TFUDR_j = \sum_{i=0}^I TF_{ij} \log(UDR_j) \quad \text{where } \Rightarrow i \in I, \Rightarrow j \in J \quad (3)$$

Equation 2 is designed to distinct upload-driven and download-driven domains while Eq. 3 calculates a value for whitelist candidate computation to select whitelist candidates. Equation 2 computes UDR (upload-download ratio) for all the visited domains of the given data set. DL is the total downloaded bytes of a given client and visited domain while UL is the total uploaded bytes of a given client and visited domain. We anticipate that a upload-driven domain will have an UDR value that is less than 1 and download-driven domains will be

greater than 1. Equation 3 multiplies the results from Eq. 1 and 2. Logarithm of UDR is design to separate upload-driven TF value from download-driven TF value. For data exfiltration detection application, we only take $TFUDR$ values that are greater than zero (which indicates download-driven domains) as the input to whitelist candidate computation component. A popular and heavy download-driven domains will expect to have a higher $TFUDR$ value, which whitelist candidate computation is most likely to select domains with a high $TFUDR$ value as whitelist candidate.

3.2 Whitelist Candidate Computation

KDE (Kernel Density Estimation) is the foundation of domain whitelist candidate selection. KDE is used to estimate probability density of sample data with kernel function and bandwidth variable to adjust distribution curve [18, 19]. We manipulate bandwidth variable to select whitelist candidates from various network feature distributions. More specifically, KDE is used as threshold value selector and any feature value surpass the threshold is selected as whitelist candidate.

From previous literature, Internet usage (domain visits) tends follow Zipf-like distribution [20–22] which means that a few domains are much more popular compare to the rest of the domains. Since whitelist candidate selection is based on popularity, we uses KDE on Zipf distribution samples to calculate a threshold value and use it as the basis for selecting domains that are qualified as whitelist candidates.

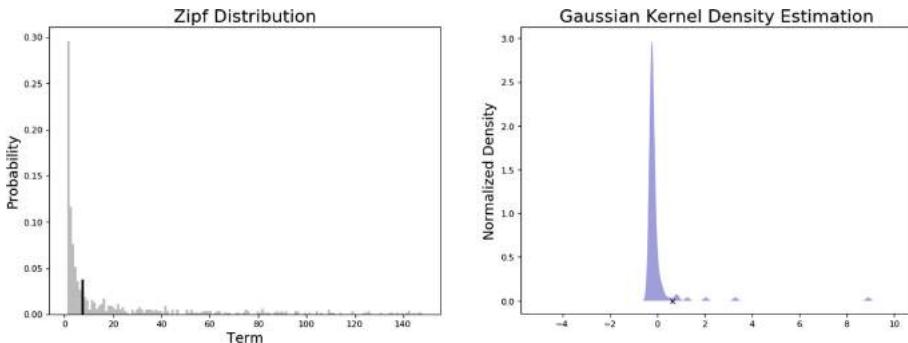


Fig. 2. Sample Zipf distribution.

Figure 2 are distribution plots using sample Zipf distribution data. Figure on the right is the plot produced by KDE function with a manipulated bandwidth. Noticed that the distribution contains several peaks of bell shape curves. Each of the bell shape curve contains a cluster of sample data, and the largest bell shape curve is the long tail sample data from Zipf distribution since the probability of the long tail sample data are very closed. If we place a mark between bell

shape curves (see the x mark in the figure), we find a threshold to divide sample data into body and long tail. This threshold value is retrieved by calculating local minimum values of the distribution. The figure of the left of Fig. 2 clearly locates the threshold value found in KDE function (see the black bar in the figure). The threshold value clearly divides body and long tail from the sample Zipf distribution and we will use the sample data in the body to serve as whitelist candidate giving its popularity.

As for bandwidth selection for KDE function, we noticed that there is no common bandwidth for different features. Therefore, we iterate through several bandwidth configurations dynamically to find a bandwidth that divides body and long tail in no more than 20–80 ratio. That is, the body population does not exceed 20% of the entire population.

After applying KDE on two sets of feature indexes (i.e. 4 indexes in feature indexes set 1 and 1 index in feature indexes set 2), 2 sets of whitelist candidates are generated. Whitelist candidate set 1 require further treatment to produce final whitelist result. Record that the 4 feature indexes of feature indexes set 1 are number of accessed clients, number of network traffic flows, total downloaded bytes, and total uploaded bytes. Candidates generated by total uploaded bytes index is used to remove duplicated candidates generated by the other 3 feature indexes giving that we are addressing data exfiltration problem. Popular domains that exercise excessive upload behavior should be removed from whitelist candidates.

3.3 Whitelist Candidate Expansion

This component leverages results from whitelist candidate computation components. There are two inputs for this component giving that two sets of feature indexes are used for candidate computation. While the two inputs are based on domain popularity on different feature indexes, popular domains often offer different types of services using numerous sub-domains. To capture these sub-domains, we extract 2nd level domains from the candidates and use it to filter out the sub-domains (domain unfolding approach [4]). As aforementioned, upload-driven domains must be excluded for data exfiltration detection. Therefore, we calculate average upload and download bytes and use it as an indication of upload-driven domains and all domains labeled as upload-driven are excluded from the whitelist. In addition, we also set up a list of well known online storage domains (e.g. Google drive, Dropbox, Onedrive, etc.) to act as whitelist filter. As mentioned, potential upload-driven domain should be excluded when addressing data exfiltration problem.

4 Evaluation

Evaluation is divided into subsections. First, we will introduce threat model that is used to carry out the experiment. It describes the approach that is used to generate anomaly network data that represents data exfiltration. Secondly, data

source and experiment set up is introduced follow by feature indexes distribution summary. Whitelist results evaluation is carried out in the fourth subsection while whitelist efficiency is evaluated in the fifth subsection. Lastly, detection performance is evaluated in the last subsection.

4.1 Threat Model

Data exfiltration can be an insider or outsider attack with a result of leaking sensitive data to adversaries. Regardless of the role of adversaries, threat model is divided into two scenarios based on sensitive data exportation. Firstly, adversaries can host a web service to store the stolen data [13]. Since the web service is used for special purpose, we can assume the web service will not receive many visits in comparison with popular web services like search engine, etc. This scenario can be detected by summarizing URL (universal resource locator) access statistics of a given network. The second scenario is the opposite, adversaries uses a popular online storage service website to export the stolen data [14–17]. In such case, URL statistics analysis will not work well. Individual host network behavior analysis can be an effective approach. We will use the introduced scenarios as evaluation input to summarize whitelist efficiency.

4.2 Data Set and Experiment Configuration

We use proxy logs from a corporate networks as the source of evaluation data. The data comprises over 10 thousand of clients that generate over 40 millions of network connection data during standard working hours (from 8am to 6pm) per day. We take 2 weeks (10 working days) worth data to train a whitelist using TOP, and 1-day data to evaluate the whitelist efficiency.

For efficiency evaluation, we use two data exfiltration detection schemes, namely naive approach and advanced approach from Marchetti et al. [3], to measure detection effectiveness with and without applying whitelist data filter. Naive data exfiltration detection approach builds network usage profile for both clients and domains. Profile contains connection count, and upload bytes statistics. Intermediate probability value is calculated to identify clients with possible data exfiltration behavior. Data exfiltration behavior is identified based on a threshold value. After possible data exfiltration behavior is filtered out, two strategies are used to rank the suspected client and domain pair. Strategy 1 ranks suspects based on total upload bytes while strategy 2 ranks suspects based on deviation degree against the profile.

We inject three different volumes of data exfiltration network traffic to the test data. Initially, 9 GB of data exfiltration network traffic (refer to incident described in [3]) were injected. However, both detection approaches are very accurate in detecting data exfiltration behavior regardless to whitelist data filtering. This is due to top upload client only consists about 1 GB of upload network traffic in the test data. For that reason, we additionally introduced 20 MB and 700 MB data exfiltration network traffic to detection efficiency evaluation.

Data exfiltration network traffic injection candidate are selected based on statistics. We select 2 domains to represent the threat model mentioned in early chapter. The first domain represents domain hosted by adversaries which is randomly selected from test data that has small amount of network traffic. The second domain represents a popular online storage network service that is has large amount of network traffic in the test data. As for client selection, we selected clients based on network traffic volume statistics. We pick 6 different clients based on connection count and upload network traffic quantile (i.e. 25%, 50%, and 75%) so that whitelist efficiency can be investigated across different network usage characteristics.

4.3 Feature Evaluation

We first analyse data features to ensure that the selected features are Zipf-like distribution, so that the features fit the assumption of TOP. Given that we have approximately over 140 thousand unique domains per day to analyse, the corresponding distribution plot is difficult to analyse visually. Therefore, we take first 75%-quantile of the distribution to summarize data distributions. Figure 3 illustrates the four selected features (number of accessed clients, number of network traffic flows, total downloaded bytes, and total uploaded bytes) for feature set one, and all four of the features are showing Zipf-like distribution.

Figure 4 illustrates *TFUDR* distribution for feature set two. We only selected the download-driven domains (i.e. *TFUDR* value that is greater than zero), since we only want download -driven domains as whitelist candidate for data exfiltration detection. Figure on the left is the first 75%-quantile distribution. Although the distribution differs from Fig. 3, it still resembles Zipf-like distribution in the plot with entire population (see the right figure on Fig. 4). The cross mark on the figure is the threshold value computed by whitelist candidate computation component where domain count is fewer than the first few clusters before domain count surges (see figure on the left). Therefore, KDE is still reliable in selecting whitelist candidates for *TFUDR* distribution.

4.4 Whitelist Evaluation

Table 1 and 2 summarize the whitelist results produced by TOP. TOP generated just over 151 thousand of FQDNs (fully qualified domain names) as whitelist for data filtering. After summarizing whitelist 1 and 2, the candidates fit the definition of whitelist in Sect. 3. Sites like [yahoo.com](#) and [google.com](#) fit the category of default page, portal, and search engine that are used widely among clients. Corporate website is also in the whitelist while [trendmicro.com](#) and [windowsupdate.com](#) are used for software updates. Lastly, [yimg.com](#) and [ads.yahoo.com](#) fit the category for commercial activities. The expand whitelist candidate component (domain unfolding strategy) worked accordingly. Sub-domains of [gstatic.com](#) took a large portion of whitelist 3 as the site is mainly responsible for delivering static contents for Google service.

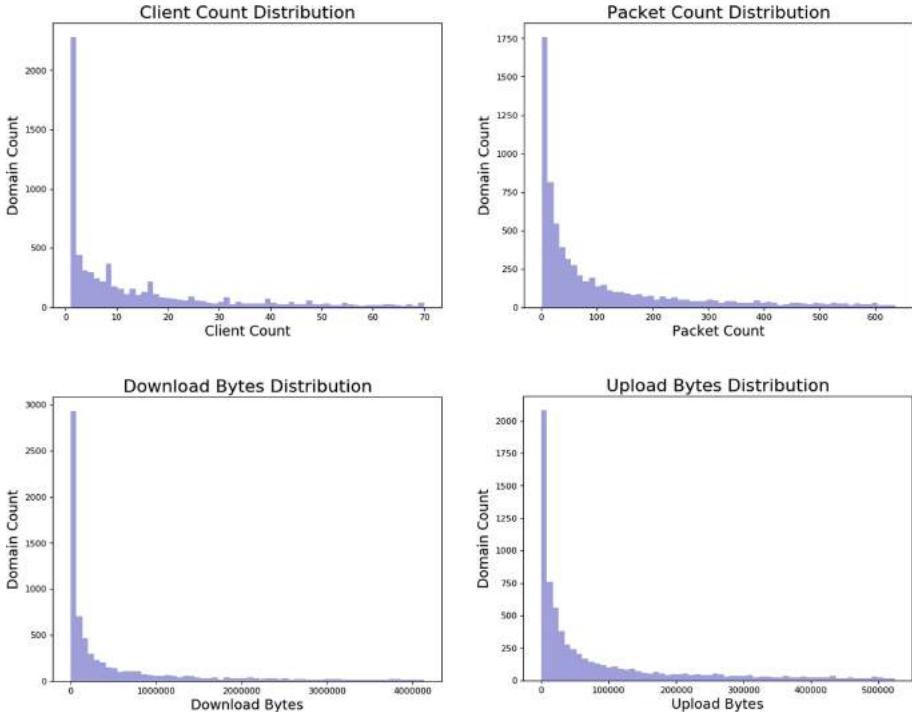


Fig. 3. Distribution of feature set one.

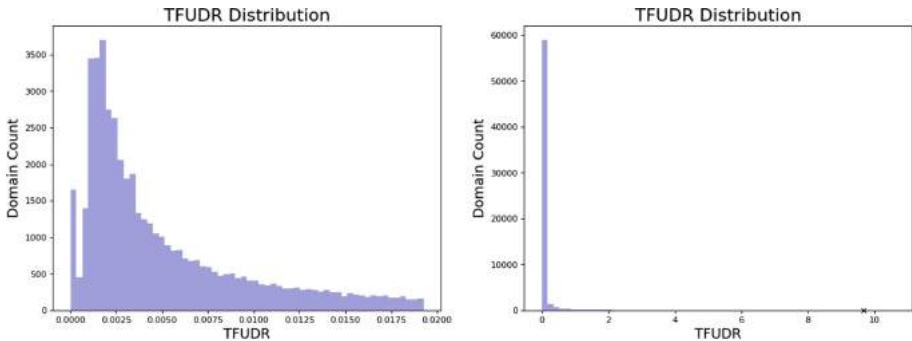


Fig. 4. Distribution of feature set two.

Noticed that a portion of candidates in whitelist 1 are not covered in whitelist 3 (see Table 2). This indicates that the excluded FQDN are not upload-driven domains, and it should not be a part of whitelist candidates. After close examination, the excluded FQDN are based on large connections and large individual client visits. They are mainly used for tracking user behavior (cookie upload network traffics) to provide commercial services such as advertisement and

recommendations (e.g. [google-analytic.com](#), [advertising.com](#), etc.). As a result, these excluded FQDNs are kept as part of the whitelist for later date exfiltration detection evaluation analysis.

Table 1. Whitelist candidate counts.

	Whitelist 1	Whitelist 2	Whitelist 3
Count	104	57	151965

Table 2. Whitelist intersection counts.

	Whitelist 1	Whitelist 2	Whitelist 3
Whitelist 1	0	30	93
Whitelist 2	30	0	57
Whitelist 3	93	57	0

Table 3 is the statistical comparison between conventional whitelist generation method based on popularity heuristics and TOP. We obtained a list of popular sites from Alexa [9, 10] and Wikipedia [23], and use it to compare filtering capability against the result from TOP. After applying domain unfolding methodology, whitelist from conventional method only generates about 19 thousand of whitelist candidates, which is only 13% of the candidates generated by TOP. Furthermore, only 6 thousand of whitelist candidates are intersected between the two approaches.

Table 4 evaluates filtering results of testing data. Filter ratio is calculated based on the number of suspected records. The sample testing data contains over 39 million of network connection records while over 27 million network connections are suspected records (where the connections have more upload bytes than download bytes). Overall, whitelist produced by TOP filters out about twice as many in contrast to conventional method.

Table 3. Conventional whitelist generation method statistics.

	Alexa global	Alexa Taiwan	Wikipedia	Domain unfold
Internet resource	50	50	50	19026
Intersected results	6	6	6	6143

Table 4. Whitelist filter efficiency statistics.

	Whitelist 1	Whitelist 2	Whitelist 3	Whitelist All	Conventional
Filter ratio	36.54%	16.61%	41.00%	50.25%	21.45%

4.5 Detection Efficiency Evaluation

Record that the data exfiltration detection approaches produce a list of suspicious candidate by ordering scores. As a result, a suspicious rank list is produced by sorting the scores in descending order. For this evaluation, we inspect the improvement of suspicious ranking after applying whitelist filtering of the experiments. After evaluating all experiment combination results (i.e. two domains, and three injected network traffic sizes), 20 MB injection network traffic of a popular web service experiment has the best improvement. All experiments that have 9 GB injection network traffic or self hosting web service have minor improvement in ranking. This is due to 9 GB injection network traffic is very suspicious already giving that the top upload traffic count is only 1 GB originally. Self hosting web service on the other hand is also very suspicious giving that the historic self hosting web service profile has a low network traffic. A surge in network traffic causes a high suspicious score. For that reason, we only presents ranking improvement result of experiment that injects 20 MB data exfiltration network traffic to a popular web service.

Table 5 and Table 6 contains suspicious ranking improvements after applying whitelist produced by TOP and conventional approach. Row is labeled by detection approach. $M1$ is the naive detection approach while $M1a$ is ranking based on total upload bytes and $M1b$ is ranking based on deviation degree against the profile. Columns is labeled by selected clients based on statistics quantile while C is connection traffic count and U is upload bytes. From Table 5, whitelist filtering shows positive results regardless to experiment configuration. Among the experiment configuration, ranking improved significantly when whitelist filtering is applied to advanced data exfiltration detection approach. Table 6 shows ranking improvement using whitelist produced by conventional method. Again, suspicious rank has improved. However, the whitelist produced by TOP shows more promising results.

4.6 Performance Evaluation

Lastly, we evaluate computation performance improvement after applying whitelist filtering to detection approaches. This evaluation is carried out on a dual Intel Xeon 2.4 GHz 10-core 2-thread (40 threads in total) machine with 160 GB memory installed. We compute two data exfiltration detection approaches with 9 GB of injected data multiple times and take the average computation time before whitelist filtering and after whitelist filtering. The first naive detection approach takes longer to compute giving that scores are calculated in network connection basis. Therefore, whitelist filtering is more effective and it

Table 5. Ranking improvement using the whitelist produced by TOP.

	C25%	C50%	C75%	U25%	U50%	U75%
M1a	2	2	2	2	2	2
M1b	5	5	5	5	5	5
M2	34	185	361	30	243	242

Table 6. Ranking improvement using the whitelist produced by conventional method.

	C25%	C50%	C75%	U25%	U50%	U75%
M1a	0	0	0	0	0	0
M1b	1	1	1	1	1	1
M2	9	91	175	11	118	109

yields 25% of computation improvement. On the other hand, the advanced detection approach calculates score on client basis which is faster to process. Whitelist filtering improves 1% of improvement giving that the entire computation only takes 0.1 s to compute.

5 Conclusion

In this paper, we presented a scheme, namely TOP, to produce whitelist to serve as data filter. A data filter that is used to filter out data noise to increase efficiency of data exfiltration detection application. Compared to conventional whitelist construction method (popularity heuristics), our scheme customizes whitelist according to study target which yields better data filtering result. According to evaluation, results from TOP can filter out twice as much data noise in comparison with the conventional whitelist generation method. Furthermore, performance evaluations also yielded better detection accuracy result and fast computation result. Although data exfiltration detection is used as the application for TOP, we believe TOP is also fit for different anomaly detection application using other Zipf-like distribution feature indexes. With minimum modification on feature indexes computation component, it is easy to apply TOP to different anomaly detection applications.

References

1. Ullah, F., Edwards, M., Ramdhany, R., Chitchyan, R., Babar, M.A., Rashid, A.: Data exfiltration: a review of external attack vectors and countermeasures. *J. Netw. Comput. Appl.* **101**, 18–54 (2018)
2. Alneyadi, S., Sithirasenan, E., Muthukkumarasamy, V.: A survey on data leakage prevention systems. *J. Netw. Comput. Appl.* **62**, 137–152 (2016)

3. Marchetti, M., Pierazzi, F., Colajanni, M., Guido, A.: Analysis of high volumes of network traffic for advanced persistent threat detection. *Comput. Netw.* **109**–2, 127–141 (2016)
4. Yen, T.F., et al.: Beehive: large-scale log analysis for detecting suspicious activity in enterprise networks. In: Proceedings of the 29th Annual Computer Security Applications Conference (ACSAC 2013), pp. 199–208. ACM, December 2013
5. Oprea, A., Li, Z., Norris, R., Bowers, K.: Made: security analytics for enterprise threat detection. In: Proceedings of the 34th Annual Computer Security Applications Conference (ACSAC 2018), pp. 124–136. ACM, December 2018
6. Hwang, T.S., Lee, T.J., Lee, Y.J.: A three-tier IDS via data mining approach. In: Proceedings of the 3rd Annual ACM Workshop on Mining network data, pp. 1–6. ACM, June 2007
7. Strayer, W.T., Walsh, R., Livadas, C., Lapsley, D.: Detecting botnets with tight command and control. In: 2006 Proceedings of 31st IEEE Conference on Local Computer Networks, pp. 195–202. IEEE, November 2006
8. Gu, G., Zhang, J., Lee, W.: BotSniffer: detecting botnet command and control channels in network traffic. In: Proceedings of the 15th Annual Network and Distributed System Security Symposium (NDSS). The Internet Society, February 2008
9. The Top 500 Sites on the Web, Alexa Internet Inc., September 2020. <https://www.alexa.com/topsites>
10. Top Sites in Taiwan, Alexa Internet Inc., September 2020. <https://www.alexa.com/topsites/countries/TW>
11. Takemori, K., Sakai, T., Nishigaki, M., Miyake, Y.: Detection of bot infected PC using destination-based IP address and domain name whitelists. *J. Inf. Process. Inf. Media Technol.* **6**(2), 649–659 (2011)
12. Kanaya, N., Tsuda, Y., Takano, Y., Inoue, D.: Byakko: automatic whitelist generation based on occurrence distribution of features of network traffic. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 3190–3199. IEEE, December 2019
13. Exfiltration Over Alternative Protocol, MITRE ATT&CK, September 2020. <https://attack.mitre.org/techniques/T1048>
14. Liu, L., De Vel, O., Han, Q.L., Zhang, J., Xiang, Y.: Detecting and preventing cyber insider threats: a survey. *IEEE Commun. Surv. Tutor.* **20**(2), 1397–1417 (2018). Second quarter
15. Homoliak, I., Toffalini, F., Guarnizo, J., Elovici, Y., Ochoa, M.: Insight into insiders and it: a survey of insider threat taxonomies, analysis, modeling, and countermeasures. *ACM Comput. Surv. (CSUR)* **52**(2), 1–40 (2019)
16. Exfiltration Over Web Service, MITRE ATT&CK, September 2020. <https://attack.mitre.org/techniques/T1567>
17. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. *IEEE Commun. Surv. Tutor.* **21**(2), 1851–1877 (2019)
18. Su, Y.H., Cho, M.C.Y., Huang, H.C.: False alert buster: an adaptive approach for NIDS false alert filtering. In: Proceedings of the 2nd International Conference on Computing and Big Data, pp. 58–62. ACM, October 2019
19. Nicolau, M., McDermott, J.: One-class classification for anomaly detection with kernel density estimation and genetic programming. In: European Conference on Genetic Programming, pp. 3–18. Springer (2016)
20. Breslau, L., Cao, P., Fan, L., Phillips, G., Shenker, S.: Web caching and Zipf-like distributions: evidence and implications. In: Proceedings of IEEE INFOCOM 1999 Conference on Computer Communications, pp. 126–134. IEEE, March 1999

21. Adamic, L.A., Huberman, B.A.: Zipf's law and the Internet. *Glottometrics* **3**(1), 143–150 (2002)
22. Guo, L., Tan, E., Chen, S., Xiao, Z., Zhang, X.: The stretched exponential distribution of internet media access patterns. In: Proceedings of the 27th ACM Symposium on Principles of Distributed Computing, pp. 283–294. ACM, August 2008
23. List of Most Popular Websites, Wikipedia, September 2020. https://en.wikipedia.org/wiki/List_of_most_popular_websites



Analyzing Co-occurrence Networks of Emojis on Twitter

Hasan Alsaif, Phil Roesch, and Salem Othman^(✉)

Wentworth Institute of Technology, Boston, MA 02115, USA
`{alsafh1, roeschp, othmans1}@wit.edu`

Abstract. Emojis have become an integral part of digital communication in recent years. They promote the expression of emotions and ideas through visual icons of people, animals, and symbols. Twitter provides a platform for discussion and commentary in real time with a diverse and global audience, making it a good candidate for studying the usage of emoji across various topics. In this paper, 296260 tweets containing emojis have been used to create undirected-weighted co-occurrence networks of emojis with respect to dating, music, US politics, Olympic Games, epidemics, in addition to a control group. For each co-occurrence network, we find related tweets by searching Twitter using associated terms. Node weight and Eigenvector centrality measures are used to compare the co-occurrence networks to one another. We find that 😊 and 🌟 are the most influential emojis in the epidemics co-occurrence network, occupying first and second place, respectively in terms of eigenvector despite not being in the top 10 emojis in terms of node weight.

Keywords: Emoji · Co-occurrence network · Graph · Social network · Twitter · Tweets · Centrality measures · Dating · Music · US politics · Olympics · Epidemics · Graph analysis

1 Introduction

Digital online communication has seen a rapid increase in the last 10 years since the introduction of smartphones, with platforms like Facebook, Instagram, and Twitter being a driving force for that change [2]. Twitter has become a source for bite sized information (280 characters) in the form of news, discussion, brand awareness, and more. The constrained length of tweets in addition to emoji enabled keyboards on smartphones compelled people to use visual expression to emphasize and convey emotions not easily expressed through text alone.

Social media has fundamentally changed the way people communicate in the 21st century, making it a daily practice for the typical person and allowing participation in a global community of shared interests such as music, sports, and US politics. Merriam Webster defines social media as: forms of electronic communication (such as websites for social networking and microblogging) through which users create online communities to share information, ideas, personal messages, and other content (such as videos).

Twitter in particular has increased the level of engagement with current events, allowing for instantaneous commentary and enriching the news cycle. This increase in engagement has made the platform attractive to advertisers, and funneled money to reach a wider audience, and propel business into the new mainstream of online presence. The new form of advertising strategy proved to be effective in making a product or services more compelling by reinforcing the brand experience in the mind of the customer. Brands also evolved their strategy to match the casual attitude of social media by replying to customers, posting memes, and using emojis just like the average person. These factors paved the way for the prevalent use of emojis on Twitter, with an estimated 300 million active users [5], making it an integral part of culture today.

Emojis are ideograms used in text-based communication and are treated by digital devices as encoded characters. They are different from emoticons, since they are pictures as opposed to typographic approximations. Emojis are part of the Unicode Consortium which allows them to be universally viewed on most digital devices. With Unicode 13.0 released in March 2020, there are 3304 represented emojis across many categories including facial expressions, everyday objects, animals, and symbols [4]. The Oxford Dictionary named ☺ the Word of the Year in 2015, and recognized the impact of emojis on popular culture. Pop culture artifacts that highlight the significance of emojis include the musical Emojiland premiered in 2016, and the Emoji Movie released in 2017. According to the study “A Global Analysis of Emoji Usage”, 19.6% of tweets from a dataset of 12,451,835 tweets contained emoji [3].

Co-occurrence networks provide a collective interconnection of terms based on their paired presence within a specified unit of text [6]. They are used to understand the relationship between people, organizations, concepts, biological organisms, and other entities represented within written material. Being able to compare the emoji co-occurrence networks to one another allows us to understand the usage of emoji across various topics. The evaluation of emoji is done using the centrality measures of node weight as well as eigenvector. Each centrality measure uncovers a property of the interconnectedness of the emoji in the network. We further discuss the details of these measures in Sect. 4. Centrality measures have been traditionally used to identify the most influential people in a social network, as well as super spreaders of disease, and in our case the most important emojis in a set of tweets. Each network centrality measures a different property of the network, which makes it difficult to settle on one universal definition of importance. Therefore, we consider the overall contribution to cohesiveness that each centrality contributes. In the remainder of this paper, we talk about related works in Sect. 2 and data collection methodology as well as search terms in Sect. 3. We elaborate on constructing the emoji co-occurrence networks in Sect. 4, and discuss the results of each network in Sect. 5. Finally, we conclude our paper in Sect. 6.

2 Related Work

The work by Illendula and Yedulla [1] created embeddings from a co-occurrence network of emojis in order to understand sentiment. Early research in sentiment analysis was focused on creating word embeddings from a large corpus of text such as Wikipedia, which is limiting for understanding the sentiment of emoji. To solve this problem, Anurag

Illendula and Manish Reddy Yedulla constructed an emoji co-occurrence obtained from tweets, and used its features to learn emoji embeddings in a low dimensional vector space to understand sentiment and emotion of emoji in the context of posts on social media. To evaluate their embeddings, they used the gold-standard dataset for sentiment analysis.

Our co-occurrence network construction method is different in that it allows nodes to have self-connections, such that when an emoji appears next to itself, its graph representation is denoted by the emoji node with an edge connected to itself. Lastly, our work is not concerned with sentiment analysis, but instead analyzes and compares co-occurrence networks about different topics with each other using node weights and eigenvector centralities.

3 Data Collection

3.1 Topics

The topics used for this research include dating, music, US politics, Olympics, epidemics, in addition to a control set. Dating is a popular topic of interest amongst young people, who are likely to use emojis with a higher frequency, thus contributing to the richness of emojis in the datasets. Music is of interest because of the size and diversity of the audience interested in that topic, as well as Twitter being a dominant platform for marketing music and allowing artists to connect with their base. US politics is a topic of interest since this research was conducted amidst the 2020 US presidential elections. It is a topic of interest since the tweets associated with it are timely, and the competitive nature of this topic allows for a constant stream of developments. In addition, the election is happening post the president's impeachment turmoil, and during the early breakout of the coronavirus pandemic, thus likely increasing the emotional sentiments of related tweets. The topic of the 2020 Tokyo Summer Olympics is also viable since Twitter is used a lot for commentating on sporting events. Epidemics are another topic of interest since this research is being conducted amidst the coronavirus outbreak. Therefore, we expect this topic to be highly influenced by coronavirus and localized to places that are being most affected by it. Lastly, a control dataset of the most common words as search term is used as a reference point for comparison with the other topics.

3.2 Collection Methodology

To collect relevant data for each of the topics discussed in Sect. 3.1, search terms are defined for each data set collection. The search terms for a given topic are intentionally neutral to minimize bias and sufficiently capture the emojis representing it during the time of collection. For example, in our Olympics dataset, we avoid using search terms of athletes and instead only pick the names of the official Summer Olympics Games. The search terms for each dataset are then queried to the Twitter API using the Python library Tweepy which returns respective tweet datasets for each topic. Although this approach does not fully eliminate bias of sentiment regarding some terms, it results in a viable starting point for studying the co-occurrence network of emojis in tweets.

3.3 Search Terms

For each of the topics, we use the search terms below and collect corresponding tweets for 3 h using the Python library Tweepy.

Dating: Arranged marriage, Autoerotic, Baby daddy, Baby momma, Bae, Break up, Bumble, Coffee meets bagel, Condom, Cuck, Cuckold, Cuffing Season, Date, Dated, Dates, Dating, deep like, Dildo, Dildos, Divorce, Dry spell, DSL, DTF, DTR, FBO, First date, Flaking, Ghosted, Ghosting, Gonorrhea, Grinder, Hinge, Hit it off, Hit off, Hitting it off, Honeymoon, Honeymoon phase, Hooked up, Hooking up, Hook up, Kiss, Kissed, Kissing, Left swipe, Left swiped, Left swipes, Love, Mail order bride, Marriage, Match, Matched, Matches, Netflix and chill, One night stand, Orgasm, Protected sex, Relationship, Right swipe, Right swiped, Right swipes, Saturdays are for the boys, Sex, Situationship, Slow Fade, Stashing, Std, Std's, Sti, Stis, Sti's, Sugar daddy, Sugar momma, Super like, Super liked, Swiped left, Swiped right, Textlationship, The clap, The love of my life, The pill, Thirst trap, Thirsty, Tinder, Valentine, Wedding.

Music: Album, Apple music, Concert, Dropping a record, Ep, Hip hop, hip-hop, Lp, Mix tape, Mumble rap, Music, Pandora, Playlist, Pop, Producer, Rap, Record, single, Song, Sound cloud, Spotify, Tidal, Tour, Track list, Trap, Vinyl.

US Politics: Democrat, Democrats, Democratic, Republican, republicans, GOP, Liberal, Liberals, Conservative, conservatives, Bernie, Bernie's, Sanders, Sanders', Sanders's, Bernie bros, Feel the bern, Buttigieg, Biden, Biden's, Warren, Warren's, Donald, Donald's, Trump, Trump's, Yang, Yang's, Yang gang, Legislation, Absentee Ballot, Ballot, ballots, Ballot Initiative, Campaign finance disclosure, Caucus, Constituent, constituents, Delegate, Delegates, Super Delegate, Super Delegates, Pledged Delegate, Pledged Delegates, Unpledged delegate, Unpledged delegates, Convention, conventions, District, districts, Election, Elector, electors, Electoral college, Electoral vote, Electoral votes, General election, Impeachment, Inauguration, Incumbent, Midterm election, nominee, Platform, Political action committee, PAC, Political party, Polling place, Polling station, Popular vote, Precinct, Precincts, Election district, Election districts, Voting district, Voting districts, Primary elections, Provisional ballot, Recall election, Recount, Referendum, Registered voter, Registered voters, Sample ballot, Special election, Super tuesday, Term, Term limit, Ticket, Town hall meeting, Town hall debate, Voter fraud, Election fraud, Voter intimidation, Voter suppression, Voting guide, Voter guide.

Olympics: Climbing, Baseball, Basketball, Gymnastics, Equestrianism, Soccer, Tennis, Track and field, Boxing, Fencing, Volleyball, Wrestling, Surfing, Golf, Artistic swimming, Ice Hockey, Modern Pentathlon, Rowing, Shooting, Swimming, Olympics weightlifting, Skateboarding, Tug of war, Water polo, Badminton, Karate, Archery, Sport Climbing, Handball, Taekwondo, Judo, Table tennis, Curling, Triathlon, Field hockey, Beach volleyball, Rhythmic gymnastics, race walking, Track cycling, Tokyo 2020, Summer Olympics.

Epidemics: Batsoup, Contagious disease, Coronavirus, Coronaviruses, Disease, Ebola, Epidemic, Epidemics, Fatality rate, Illness, Influenza, Malaria, Outbreak, pandemic, Plague, Plagues, Quarantine, Swine flu, Wou flu, Wu flu, Wuhan.

4 Constructing Emoji Co-occurrence Network

The emoji co-occurrence networks are constructed from 296260 tweets containing emoji with respect to the topics of dating, music, US politics, Olympic Games, epidemics, in addition to a control group. For each topic, the corresponding tweets are collected over a period of three hours by searching Twitter using related terms discussed in Sect. 3.3. The tweets were collected on February 29th, 2020. Each distinct emoji in a tweet generates a node of m edges where m is the number of distinct emojis in the tweet. The weight of a node signifies the number of occurrences of that node, while the weight of an edge signifies the number of co-occurrences of nodes incident to the edge. If a node appears more than once in a tweet, a self-looping edge is added signifying the co-occurrence of the node to itself. Figure 1 shows the construction of a co-occurrence network from a sample tweet. The co-occurrence network gains more edge connections as well as nodes with additional tweets. Figure 2 shows the growth of the co-occurrence network from a sample of one tweet in Fig. 1 to a sample of two tweets. In the case of the co-occurrence network in Fig. 2, because the 🐚 and 🌎 emojis already exist, their node weights as well as edges are incremented according to the new additional tweet. However, the 🏔 node is not previously present in the network, and is thus added with incident edges to 🐚 and 🌎.

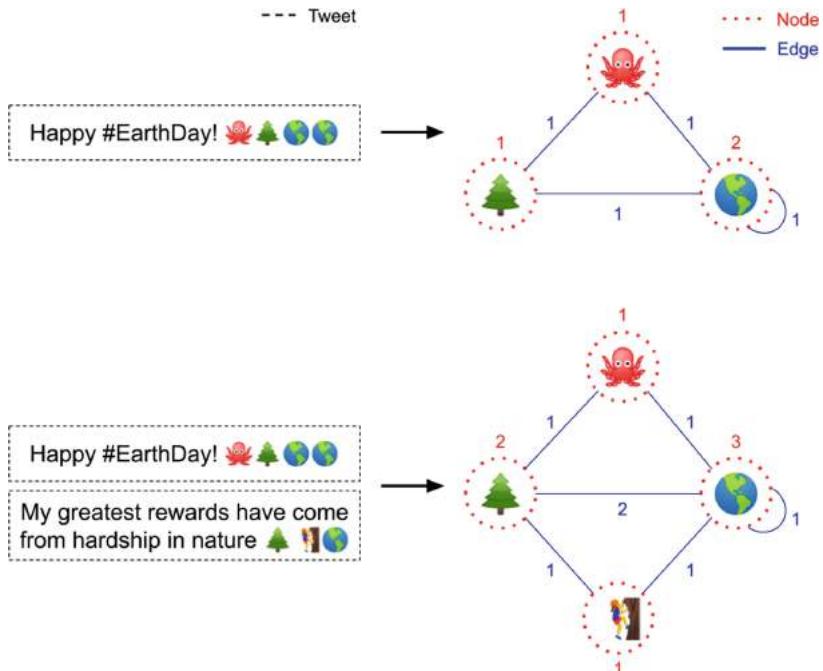


Fig. 1. Co-occurrence Network Construction Example.

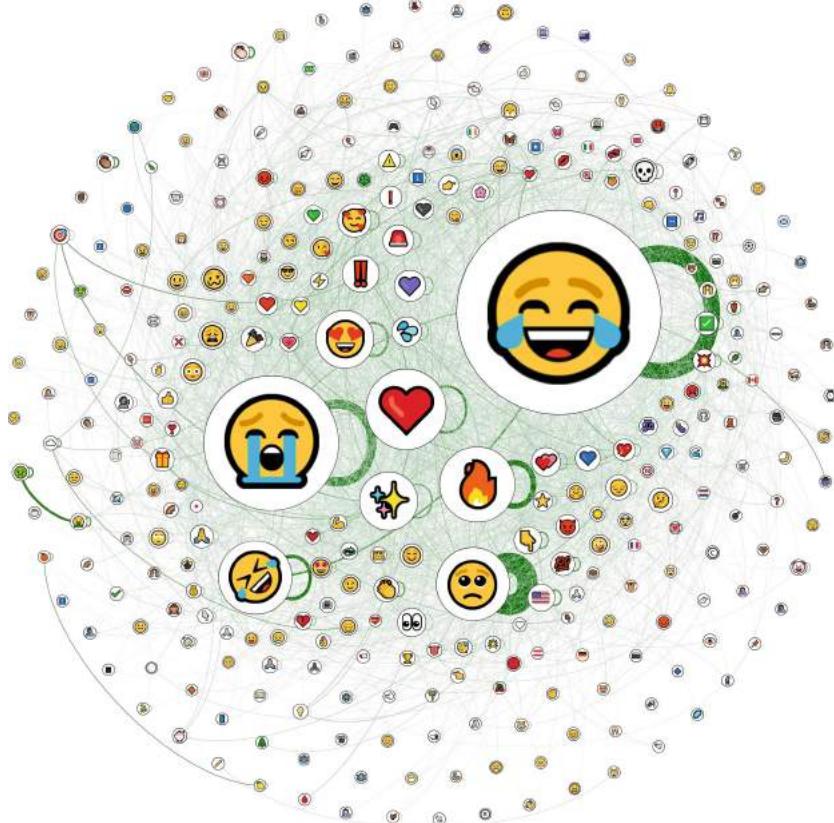


Fig. 2. The Co-occurrence Network Created from the Control Tweets Dataset. The Graph is Filtered to Only Include Nodes with a Weight Greater than 100, And Edges Weights Greater than 10.

5 Results

5.1 Eigenvectors

Eigenvector centrality plays a significant role in understanding the structure of our graph since it allows us to find the most influential node [7] or emoji. Figure 3 below is the mean of the eigenvectors from the five topic datasets sorted by descending order.

Top Emojis by Eigenvector

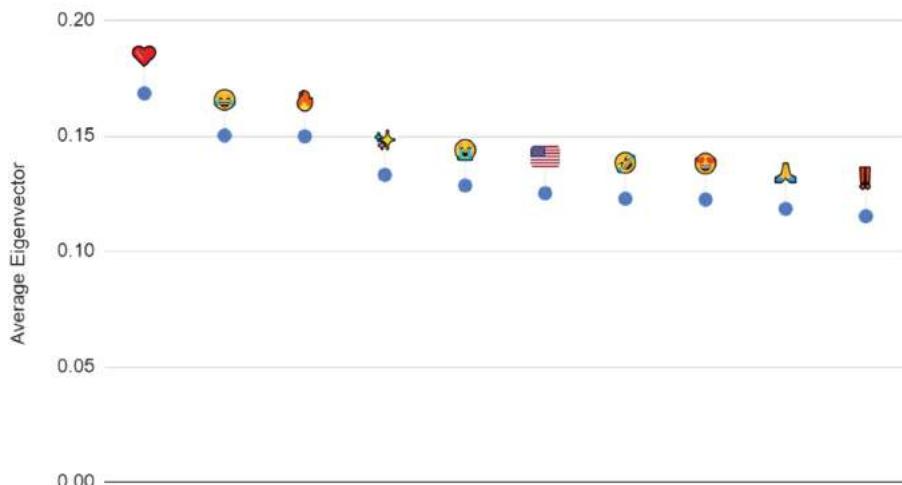


Fig. 3. Emojis from the Dating, Music, US Politics, Olympics, and Epidemics Graphs Sorted by their Average Eigenvector Value.

We see that the ❤️ has the highest eigenvector. However, taking the intersection of the top eigenvectors from each dataset gives us slightly different results. The example below shows the intersection of the top six emojis for three dataset resulting in the face with tears of joy emoji 😢.

One of the first questions we ask is, how do we compare the co-occurrence networks to each other? We try to answer this question by finding the intersection of top emojis in terms of eigenvector as illustrated by the venn diagram (Fig. 4). However, finding the first intersection emoji “😢” for all five datasets requires that we expand to include the top 15 emojis from each dataset. We further increase the number of top emojis in our comparison to find more emojis in the intersection of our datasets displayed in Fig. 5. The numbers on the y-axis are calculated by averaging the eigenvector values of each emoji across all datasets, and the x-axis denotes the number of top emojis required to find the intersecting emoji.

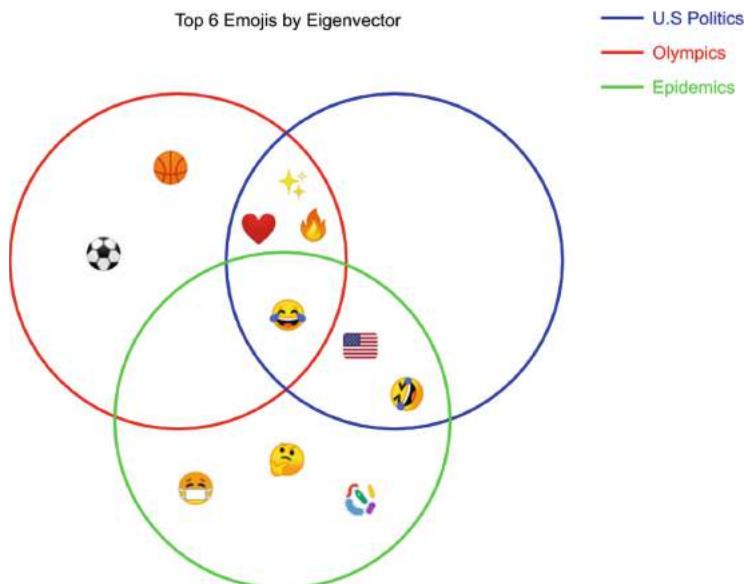


Fig. 4. Venn Diagram Displaying the Top 6 Emojis from the US Politics, Olympics, and Epidemics Co-Occurrence Networks.

Top Emoji Set Intersections by Eigenvector

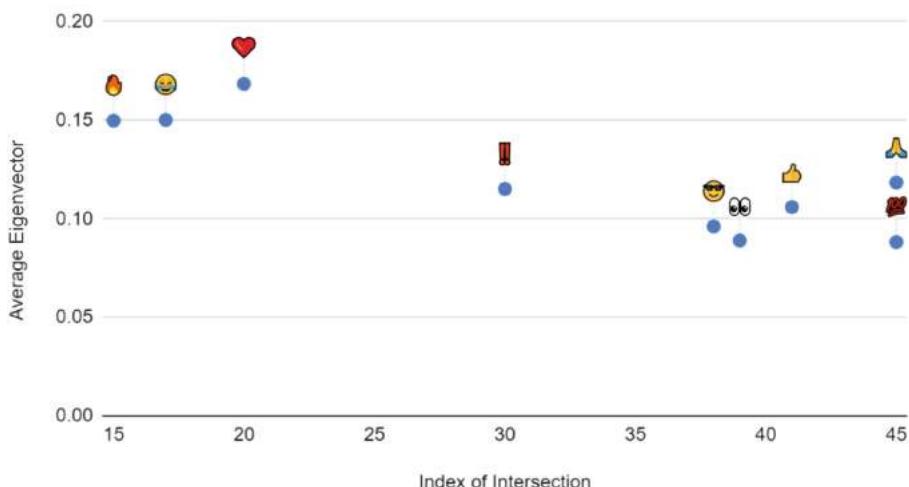


Fig. 5. The Intersection of the Top Emojis from the Dating, Music, US Politics, Olympics, and Epidemics Graphs. The y-Value of the Emojis is given by their Average Eigenvector Value Across Graphs.

5.2 Dating

The weight of ❤️ in this dataset is the most frequent (Fig. 7), and is quite fitting for tweets related to dating. For the eigenvector centrality however, which is an indicator of nodes connecting other nodes in the graph to each other, ✨ occupies third place after the heart emoji (Fig. 6). Although ✨ is the ninth most frequent emoji in the dataset, about four times less frequent than ❤️, it plays a significant role in connecting other emojis to each other in the network.

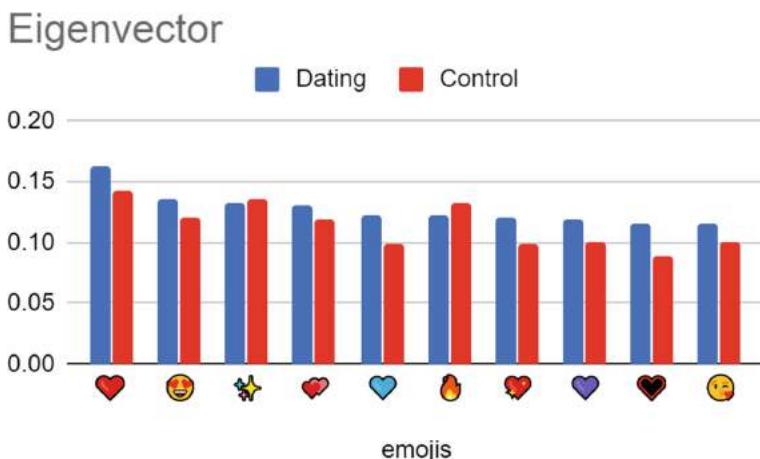


Fig. 6. The Top 10 Emojis by Eigenvector in the Dating Graph.

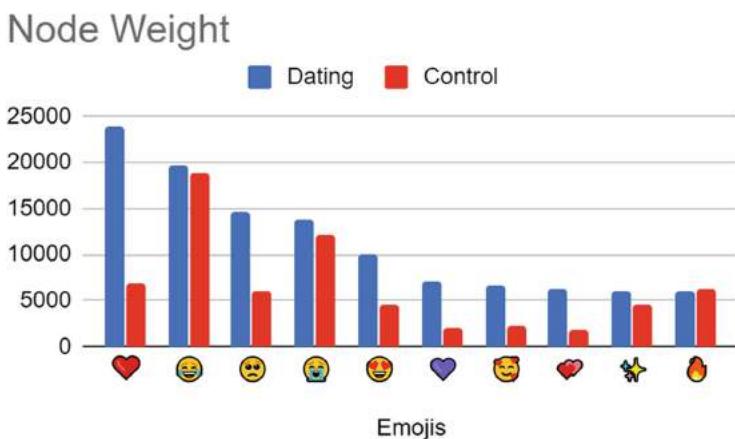


Fig. 7. The Top 10 Emojis by Node Weight in the Dating Graph.

5.3 Music

For the music dataset, the top three emojis in terms of frequency are the 🎵, 💃, and ❤️ respectively (Fig. 9). In terms of eigenvector centrality, the 🎵, ❤️ and ✨ emojis take the top three places respectively. However, the fourth place in terms of eigenvectors is (Fig. 8), which did not appear in the top ten nodes in terms of frequency.

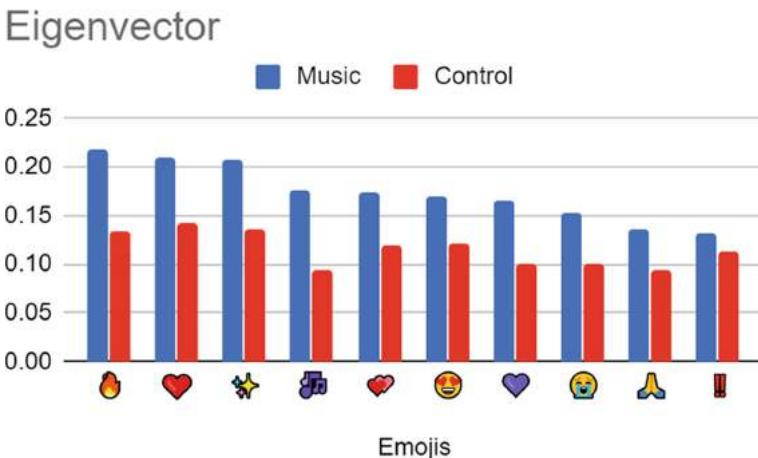


Fig. 8. The Top 10 Emojis by Eigenvector in the Music Graph.

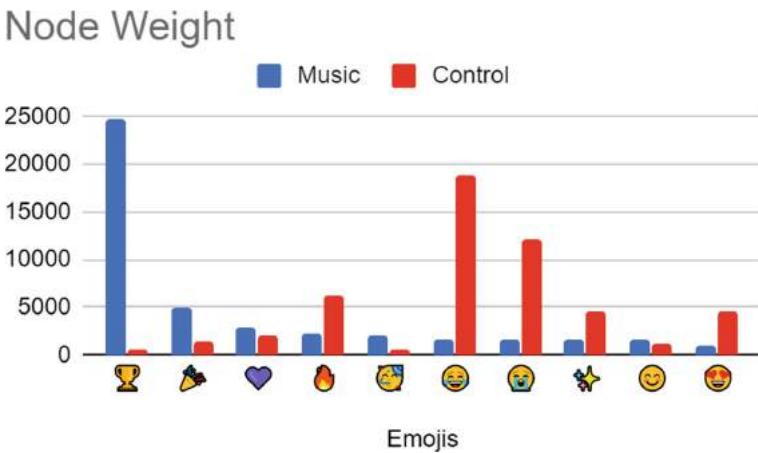


Fig. 9. The Top 10 Emojis by Node Weight in the Music Graph.

5.4 US Politics

For the US politics dataset, the most popular nodes include 🗳️, 🏛️, and 🚩 (Fig. 11). 🚩 as well as 🚨 emojis are also quite popular. The fire emoji occupies sixth place, while

♡ occupies ninth place in terms of frequency. Surprisingly, although ♡ is ninth in terms of frequency, it occupies sixth place for eigenvector centrality and US occupies first (Fig. 10). 🌟 comes is in sixth place, while ✨ comes in fifth in terms of eigenvector centrality.

Eigenvector

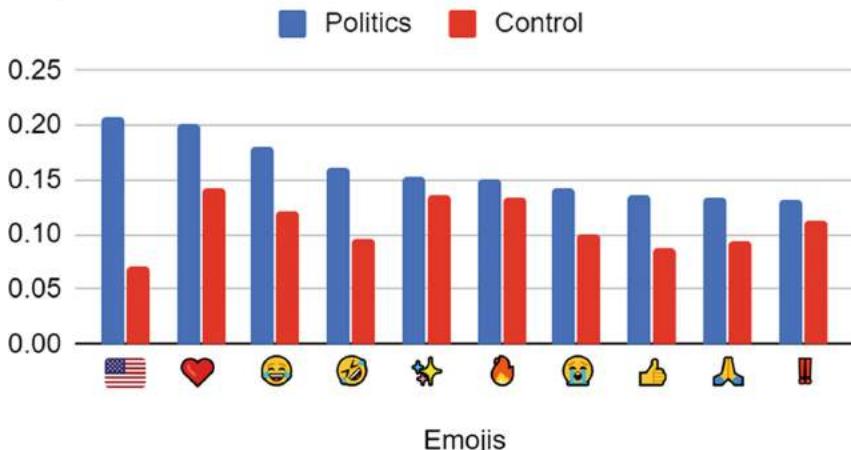


Fig. 10. The Top 10 Emojis by Eigenvector in the US Politics Graph.

Node Weight

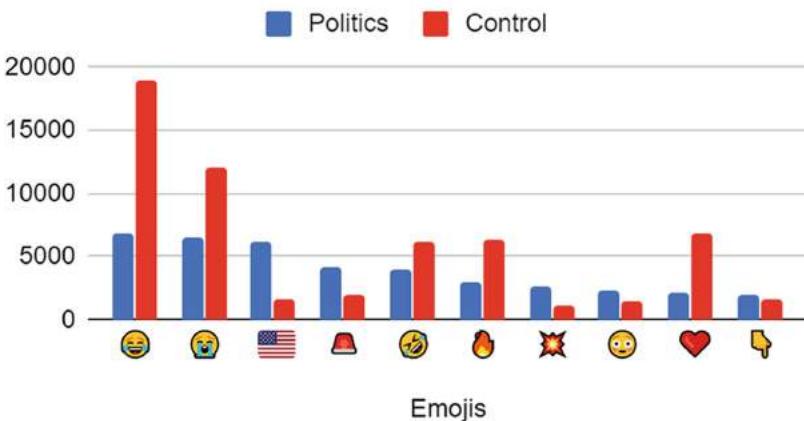


Fig. 11. The Top 10 Emojis by Node Weight in the US Politics Graph.

5.5 Olympics

For the Olympics dataset, the most popular emojis include 😊, ✓, 🏆, , and surprisingly (Fig. 13). In terms of eigenvalue, ❤️ occupies first place, ✨ second place, third place (Fig. 12). However, 🏋️‍♂️ occupies third place in eigenvector centrality, while ⚽ occupies fifth place.

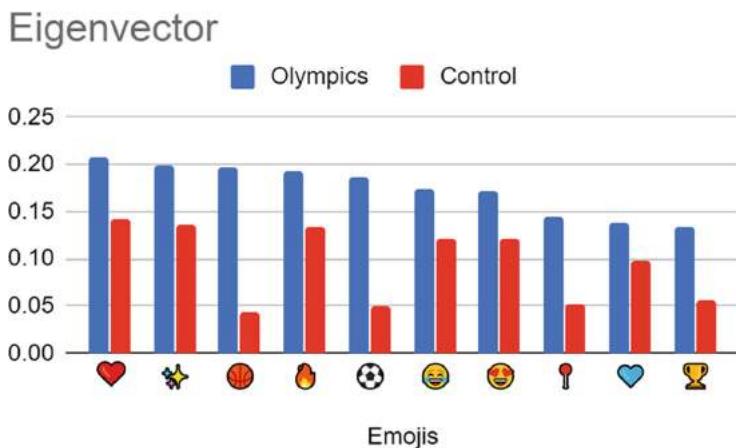


Fig. 12. The Top 10 Emojis by Eigenvector in the Olympics Graph.

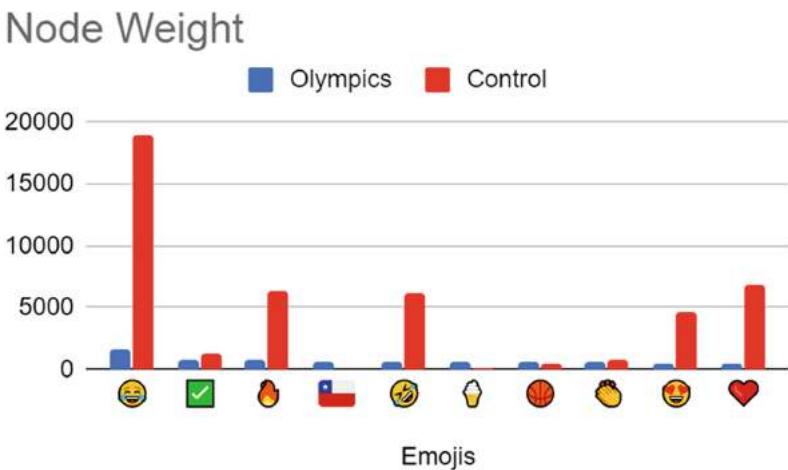


Fig. 13. The Top 10 Emojis by Node Weight in the Olympics Graph.

5.6 Epidemics

For the epidemics dataset, the and emojis stand out and occupy fourth and fifth place, respectively in terms of node frequency (Fig. 15). They are followed by , , , and signifying countries impacted by the coronavirus. Surprisingly, the top two in eigenvector centrality emojis are and , respectively (Fig. 14), which both did not appear in the top ten nodes in terms of frequency. Emojis, such as , , and which usually appear high on the eigenvector centrality have relatively low scores in the epidemics dataset. An interesting finding is that is at number thirty in terms of node frequency with a weight of 1229 compared to at number one with a weight of 6118, it appears to be number four in terms of eigenvector centrality and the only country to make it to the top 10.

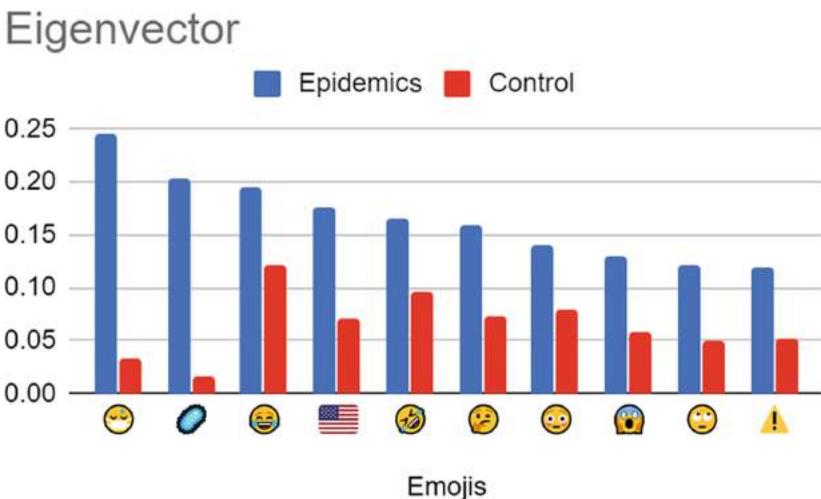


Fig. 14. The Top 10 Emojis by Eigenvector in the Epidemics Graph.

5.7 Control Dataset

Once we analyze the centralities of the control dataset, we can determine a handful of emojis are most responsible for connecting the nodes of the network to each other. is the most frequent with about twice as many as at second place and three times as many as at third place. When we compare the other datasets to the control, we can observe that some emojis appear frequently in the network such as as well as , indicating that users who use these emojis use them frequently in any given situation. Another observation is that some emojis may not appear as frequently, but have an overall more significant role such as , , and emojis with high eigenvector centralities.

Node Weight

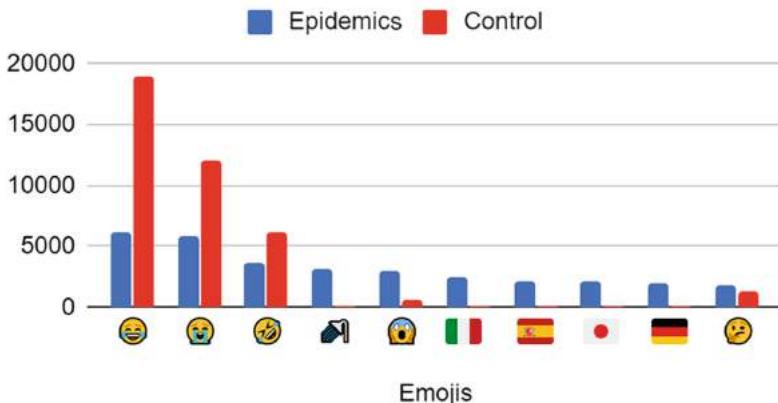


Fig. 15. The Top 10 Emojis by Node Weight in the Epidemic Graph.

5.8 Data

The tables below present data regarding the topics discussed in Sect. 3.1. The data includes the number of tweets Table 1 used to construct the co-occurrence network, the top emojis by node weight Table 2, as well as the top emojis by eigenvector for each of the given topics Table 3.

Table 1. Number of Tweets with Emojis in the Control, Dating, Music, US Politics, Olympics, Epidemics Datasets.

Number of Tweets with Emojis							
Control	Dating	Music	US Politics	Olympics	Epidemics	Total	
68293	125426	59278	15193	7002	21068	296260	

Table 2. Top 10 Emojis by Node Weight in the Control, Dating, Music, US Politics, Olympics, and Epidemics Graphs.

Top 10 Emojis by Node Weight						
Control	Dating	Music	US Politics	Olympics	Epidemics	
😂	18900	❤️	23936	🏆	24758	😂
🇺🇸	12106	😎	19673	🎮	4945	🇺🇸
❤️	6808	😢	14724	❤️	2777	🇺🇸
🎧	6317	🌐	13749	🎧	2320	💻
🚫	6136	☕️	10062	🎮	2137	🚫
愀	6098	❤️	7102	😊	1634	␣
✨	4647	🌍	6599	🌐	1559	💥
😍	4615	❤️	6299	✨	1539	😳
❗	2551	✨	6087	😊	1522	❤️
🎭	2227	🎧	6067	😊	957	👉
						1959
						Hearth
						495
						💡
						1811

Table 3. Top 10 Emojis by Eigenvector in the Control, Dating, Music, US Politics, Olympics, and Epidemics Graphs.

Top 10 Emojis by Eigenvector						
Control	Dating	Music	US Politics	Olympics	Epidemics	
❤️	.1424	❤️	.1617	🎧	.2186	🇺🇸
✨	.1362	🌍	.1355	❤️	.2084	❤️
🎧	.1329	✨	.1330	✨	.2066	😊
😍	.1213	❤️	.1310	🎵	.1762	🚫
😂	.1206	❤️	.1219	❤️	.1731	✨
❤️	.1187	🎧	.1213	😍	.1693	🎧
❗	.1136	❤️	.1210	❤️	.1653	🇺🇸
🎭	.1071	❤️	.1189	🏆	.1526	👍
👀	.1045	❤️	.1156	🎮	.1366	👍
⚡	.1016	⚡	.1153	❗	.1311	❗
						.1326
						🏆
						.1347
						⚠️
						.1186

6 Conclusion

From the results of our study, we concluded that some emojis such as ☺, ✨ and ❤ have a more important role in influencing the network, as measured by their eigenvectors, and this role appears to be universal across the five datasets as well as the control. However in some cases, non-common or particularly universal emojis, such as 😊 and 🌐 in the epidemics graph become outliers. In the epidemics dataset, the rapid global spread of coronavirus and recommendation of health experts to wear masks were contributing factors for the 🌐 and emojis to become the most important in terms of their eigenvector measurements. These nodes have high eigenvectors and appear to gain influence even with a relatively small frequency compared to the nodes with the highest frequency in the graph. In other cases, some nodes can have the converse property, with a high frequency but very little influence in terms of eigenvectors such as the 🎵 in the music graph. This suggests a non-deterministic relationship between the frequency of emojis in the graph and their eigenvectors. The eigenvector of emojis is of course dependent on the news and search terms of the data set at the time of collection. This lack of correlation also appears to exist amongst universally popular emojis in terms of frequency as well as eigenvectors. 😊 for example is widely popular and holds the second highest average eigenvector, however it does not make the top ten eigenvectors in the music graph even though it places second highest in terms of frequency.

7 Future Work

We can directly expand this study by collecting larger data at a different time to see how the frequency of emojis change, as well as collecting data from sources other than Twitter. This will perhaps show more insight as to how the medium affects people's usage of emojis, and also show which emojis are commonly used across different mediums for the same topic and over time. Our co-occurrence network can also be used for a user predictive model by analyzing emojis that are clustered together. In a similar manner, Google has released Emoji Kitchen, a Gboard feature that allows users to combine multiple emojis into a new singular emoji in February 2020 [8]. The network can be expanded by including the given text for prediction which creates a co-occurrence of emojis and words.

References

1. Illendula, A., Yedulla, M.R.: Learning emoji embeddings using emoji co-occurrence network graph. In: 1st International Workshop on Emoji Understanding and Applications, ICWSM (2018)
2. Edosomwan, S., Prakasan, S.K., Kouame, D., Watson, J., Seymour, T.: The history of social media and its impact on business. *J. Appl. Manag. Entrepreneurship* **16**(3), 79–91 (2011)
3. Ljubešić, N., Fišer, D.: A global analysis of emoji usage. In: Proceedings of the 10th Web as Corpus Workshop (WAC-X) and the EmpiriST Shared Task, pp. 82–89 (2016)
4. Emoji Counts, v13.0. Unicode.org. <https://www.unicode.org/emoji/charts-13.0/emoji-counts.html>. Accessed 8 Sept 2020

5. Number of monthly active Twitter users worldwide from 1st quarter 2010 to 1st quarter 2019. <https://www.statista.com/statistics/282087/number-of-monthly-active-twitter-users/>. Accessed 8 Sept 2020
6. Ozdikis, O., Senkul, P., Oguztuzun, H.: Semantic expansion of tweet contents for enhanced event detection in Twitter. In: ACM International Conference on Advances in Social Network Analysis and Mining 2012, Istanbul, pp. 20–24. IEEE (2012)
7. Bonacich, P.: Some unique properties of eigenvector centrality. *Soc. Netw.* **29**(4), 555–564 (2007)
8. Feeling All the Feels, There's an Emoji for That. <https://blog.google/products/android/feeling-all-the-feels-theres-an-emoji-sticker-for-that/>. Accessed 11 Jan 2021



Development of an Oceanographic Databank Based on Ontological Interactive Documents

Oleksandr Stryzhak¹, Vitalii Prykhodniuk¹, Maryna Popova^{1(✉)},
Maksym Nadutenko², Svitlana Haiko³, and Roman Chepkov⁴

¹ National Center “Junior Academy of Sciences of Ukraine”, 38-44, Dehtiarivska Street,
Kyiv 04119, Ukraine

² Ukrainian Lingua-Information Fund of NAS of Ukraine, 3, Holosiivskyi Avenue,
Kyiv 03039, Ukraine

³ Institute of Telecommunications and Global Information Space of NAS of Ukraine, 13,
Chokolivs'kyi Blvd., Kyiv 03186, Ukraine

⁴ Scientific and Research Institute of Geodesy and Cartography, 69, Velyka Vasyl'kivs'ka
Street, Kyiv 03150, Ukraine

Abstract. The article proposes an approach to development of an oceanographic databank based on the formation of an ontological representation of information arrays and accessing them with an ontology-defined interface. For processing large amounts of oceanographic data located in disparate archives and databases (including those represented as natural language documents) recursive reduction method is proposed. For providing interactive access to information and services ontological interactive document is suggested. Its core element, called ontological presentation template, ensures the most effective work of the experts with the information and allows them to change system's structure, composition and functions with minimal time. Control ontologies as a mechanism for defining the configuration of the natural system can be used. For increasing the efficiency of the natural system by optimizing the information transferring process ontological integration descriptors are suggested. Approach to oceanographic data processing that includes recursive reduction method for textual scientific reports and measurement result files processing is presented. Oceanographic databank software system with ontology-based user interface and GIS application viewing mode is presented.

Keywords: Oceanology · Oceanography · Data processing · Natural language processing · Word processing · Data storage systems · Databank · Ontology · Interactive document

1 Introduction

Information support serves as the basis for decision-making in the field of maritime activities at all levels. Domestic and foreign experience in the field of retrieving and using environmental information shows that the greatest results can be obtained if all

types of activities for the collection, permanent storage, processing and exchange of oceanographic data are carried out within the framework of a single information system.

Such oceanographical systems have already been created or are being created in many countries of the world. For Ukraine, until recently (2014) processes for collecting, storing and exchanging oceanographical data were carried by organizations, located in the Crimea. After their loss, such processes were mostly carried independently by different organizations, and direct information exchange between them mostly ceased. As the result, large volumes of vital data, currently present in Ukraine, are usually stored in disparate archives and databases and inaccessible to a wide range of consumers and interested professionals. Those data include hydrological, hydrophysical, hydrochemical, hydrometeorological, geological-geophysical measurements of the marine environment, data on living and non-living resources, storm disturbance, seismic characteristics of the Azov-Black Sea and the Black Sea basin.

Nowadays the issue is being raised about reorganization of these processes with creation of the specialized databank for Academy of Sciences of Ukraine to serve as central data storage and exchange hub. Creation of such databank requires powerful IT-infrastructure with ability to process large volumes of weakly structured data automatically. Such infrastructure should also provide tools to extract measurements from natural-language texts, as some of the data was lost and has to be recovered from secondary sources (including scientific reports, journal articles etc.).

An approach for creating such databank is proposed, which is based on method of recursive reduction [1–3] for structuring weakly structured and unstructured data (including natural-language texts) and on creation of ontological interactive documents for further processing, displaying and exchanging resulting structured data.

The structure of the paper is the following. Section 2 describes current state of oceanographical data exchange procedures and challenges met by Ukraine in this field. Section 3 describes recursive reduction method used to structure weakly structured arrays of data (including natural-language texts). In Sect. 4 the description of the ontological interactive document is given, which is a kind of information system that uses ontologies for storing information to be processed and as a configuration for defining available functions. In Sect. 5, the input array of oceanographic data and approach for its processing are described. Section 6 presents oceanographic databank software system.

2 State of the Art

Collecting and exchanging oceanographical data is a very important task, for which large networks of various systems and institutions are deployed worldwide. Many of them are deployed under various international programs, like The European Marine Observation and Data Network (EMODnet) and Copernicus Marine Environment Monitoring Service (CMEMS).

This process is governed by Intergovernmental Oceanographic Commission, which states, that all oceanographical data should be generally routed through National Oceanographic Centers through radio communications or Internet [4], which requires corresponding infrastructure to be deployed in specific country.

Ukraine needs deployment of such infrastructure, which requires substantial time and funding. Currently most of oceanographical data is collected during expeditions, and

following processes (like verifying and error correcting) is performed either manually, or with basic semi-automatic tools. The collected data are stored as arrays of weakly structured information or as parts of scientific reports (which a mostly natural language texts) [5]. Such data cannot be effectively used and exchanging them with standard procedures requires large amounts of manual labor to structure them properly.

Presented approach to creation of oceanographical databank eliminates most of the manual labor and streamlines data collection and exchange processes. Such databank can become a basis of automatic data collection infrastructure when it will be deployed.

3 The Method of Recursive Reduction

Large amount of available in Ukraine oceanographical data located in disparate archives and databases, often in formats, not optimized for automatic processing. For accessing large part of these data natural language processing is required. For accomplishing this task a method of recursive reduction [1, 2, 6, 7] is proposed. The method of recursive reduction is used to structure weakly structured and unstructured documents. The results of using this method represented as ontologies [8–10], which can be viewed as ordered triples (1).

$$O = \langle X, R, F \rangle \quad (1)$$

where X is the set of objects of the subject area, R is the set of relations between objects, F is the set of interpretation functions of X and R .

The structuring of a certain natural-language text can be represented as a kind of transformation (2).

$$F_{str} : T^T \rightarrow O \quad (2)$$

Any natural language text is represented by a set of lexemes L , on which a certain relation of the preceding [1, 2, 11–13] is defined. This relation turns L into a linearly ordered set. In addition, the text can be represented as a sequence of sentences (3), on which the same relation of preceding is defined.

$$T^T = \{S_1 \prec S_2 \prec \dots \prec S_{n_s}\} \quad (3)$$

where n_s is the total number of sentences in the text.

Each sentence S_i , in turn, is represented by some subset of lexemes (4).

$$L_{S_i} = \{l_{i1} \prec l_{i2} \prec \dots \prec l_{in_i}\} \quad (4)$$

where n_i is the number of lexemes in the i -th sentence. Each lexeme, in turn, has a structure (5).

$$l_{ij} = \langle l_{ij}^T, P_{ij} \rangle \quad (5)$$

where l_{ij}^T is the textual representation of the lexemes l_{ij} , P_{ij} are characteristics of l_{ij} .

The lexeme can be connected with other lexemes by relations (6).

$$r_{sn} = \langle l^1, l^2, k \rangle \quad (6)$$

where l^1, l^2 are lexemes between which there is a relation, k is a type of relation.

Thus, a directed graph representing the basic structure of a natural-language text has the form (7).

$$T_{sn} = \langle L, R_{sn} \rangle \quad (7)$$

3.1 Reduction Operator

The first step of the transformation (2) is forming a structure (7) based on the input text. This process is carried out using lexical analysis. Then, the reduction operator (8) is applied recursively to this structure. The reduction operator is described in terms of λ -expressions [1, 2, 14, 15], and, in turn, is a combination of three other operators applied sequentially.

$$F_{rd} = F_x \circ F_{smr} \circ F_{ct} \quad (8)$$

where F_x – an object identification operator, F_{smr} – a relation identification operator, F_{ct} – a context identification operator.

The operator (8) applied recursively, for which the fixed-point operator [14] is used (9).

$$Y = \lambda f.(\lambda x.f(xx))\lambda x.f(xx) \quad (9)$$

The recursion exit condition is verified by the means of an auxiliary function (10).

$$F' = \lambda fx.\begin{cases} fF_{rd}x, & F_{rd}x \neq x \\ x, & F_{rd}x = x \end{cases} \quad (10)$$

Each of the elements of the reduction operator (8), in turn, is formed by rules of the form (11).

$$g = \langle f_{ap}^g, f_{tr}^g \rangle \quad (11)$$

where f_{ap}^g is the applicability function, which determines whether a rule can be applied to a specific set of input information, f_{tr}^g is the conversion function, which defines the conversion of input information.

The transformation specified by rule g has the form (12).

$$F_g(x) = \begin{cases} f_{tr}^g(x), & f_{ap}^g(x) \\ x, & \neg f_{ap}^g(x) \end{cases} \quad (12)$$

3.2 Applicability Function

Each applicability function is a lambda term of the form (13).

$$f_{ap} = (\lambda l_1, \dots l_n. t_l(\bar{l}) \& t_r(\bar{l})) x_1, \dots x_n \quad (13)$$

where l_i is a variable taking values on the set of lexemes, x_i is a function argument that defines the value l_i , t_l – lexeme identification function, t_r – relation identification function.

The applicability function imposes a condition [1, 6, 16] on the oriented graph, formed by a subset of lexemes, belonging to a single sentence (4) and relations between those lexemes. In this case, the condition is imposed sequentially on each lexeme and on each of the relations between them. Imposed conditions are defined as sets of predicates.

The lexeme identification function is defined by a set of lexeme identification predicates and has the form (14). Predicates are designed to test various aspects of a particular lexeme from an input sequence.

The main type of such predicates is the keyword-checking predicates (15). Other standard types are characteristics-checking predicate (16) and zero predicate (17). A zero predicate is used in a situation when irregular lexemes are expected. Those lexemes can be absent in keyword dictionaries, and there is a probability of incorrect identification of their characteristics during lexical analysis. It is also possible to use specialized types of predicates within specific tasks, for example, regular expression predicates.

$$t_l(l_1 \dots l_n) = c_1(l_1) \& c_2(l_2) \& \dots c_n(l_n) \quad (14)$$

where c_i are lexeme identification predicates.

$$c(l) = \begin{cases} 1, & l^T \in L_c \\ 0, & l^T \notin L_c \end{cases} \quad (15)$$

where l^T – textual representation of the lexeme l , L_c – keyword dictionary checked by predicate c

$$c(l) = \begin{cases} 1, & p_c \in P_l \\ 0, & p_c \notin P_l \end{cases} \quad (16)$$

where P_l – the set of characteristics of the lexeme l , p_c – a characteristic, checked by predicate c .

$$c(l) = 1 \quad (17)$$

The relation identification function has the form (18) and is determined by the set of relation identification predicates. Predicates have a form (19) and are defined by the type of connection that must exist between certain lexemes in the input sequence. In general, not all lexemes are connected with relations, so most predicates are zero predicates (20).

$$t_r(l_1 \dots l_n) = \begin{cases} 1, & \forall i, j \in [1..n], i \neq j, r_{ij}(l_i, l_j) \\ 0, & \exists i, j \in [1..n], i \neq j, \neg r_{ij}(l_i, l_j) \end{cases} \quad (18)$$

where $r_{ij}(l_i, l_j)$ – relation identification predicates.

$$r(l_1, l_2) = \begin{cases} 1, & \langle l_1, l_2, k_r \rangle \in R_{sn} \\ 0, & \langle l_1, l_2, k_r \rangle \notin R_{sn} \end{cases} \quad (19)$$

where k_r – the type of relation that is checked by the predicate.

$$r(l_1, l_2) = 1 \quad (20)$$

3.3 Transformation Function

The transformation function f_{tr}^g is intended for the formation of objects and relations in resulting ontology (1), as well as contexts, which are used to define interpretation functions. For each of the components of the reduction operator (8), the transformation functions contained in the corresponding rules will have a specific structure. However, all of them are based on the name creation function (21), which, in turn, uses the formatting functions on textual representations of individual lexemes. Formatting functions can be very different depending on the task being performed.

$$N(l_1, \dots, l_n) = \sum_{i=1..n} f_i^N(l_i^T) \quad (21)$$

where f_i^N – text formatting function, l_i^T – text representation of lexeme l_i .

The main operations that can be performed during formatting are:

- Zero operation, i.e. use text representation without changes;
- Skipping a lexeme, i.e. not including its textual representation in the name;
- Normalization of the lexeme, i.e. using the corresponding word in the nominative case of the singular (for lexemes representing the words);
- Normalization of the number format (for lexemes representing numbers);
- Normalization of geospatial information – formatting various types of coordinates, if necessary – with conversion between coordinate systems;
- Normalization of dates – parsing textual representations of dates and bringing them to a single format.

The transformation functions for the various stages of recursive reduction have the following structure:

The object creation function is used in rules that define the object identification operator F_x and has the form (22).

$$f_{tr}^g(l_1, \dots, l_n) = X(N(l_1, \dots, l_n)) \quad (22)$$

where X is the operation of creating an object with a given name.

The relation creation function is more complex (23). These functions are used in rules, that define relation identifying operator F_{smr} . The job of this function is to determine two objects and create a relation between them.

$$f_{tr}^g(l_1, \dots, l_n) = R(X(N(l_1, \dots, l_m)), X(N(l_{m+1}, \dots, l_n))) \quad (23)$$

where X – the operation of creating (or selecting) an object with a given name, R – the operation of creating a relation between two given objects, m – an index that indicates the boundary between object names in the input lexeme sequence.

The attribute creation function has the form (24). This kind of function often requires specialized procedures for determining the object to which the attribute belongs. In the general case, this requires preserving the text processing state in a certain way (for example, remembering the last identified object).

$$f_{tr}^g(l_1, \dots, l_n) = A(N(l_1, \dots, l_m), N(l_m, \dots, l_n)) \quad (24)$$

where A – the operation of creating an attribute by its name and value, m – index indicating the boundary between the name and value.

4 Ontological Interactive Documents

Ontological interactive document [1, 2, 17, 18] is a kind of information system, that uses ontologies as a source of information for both displaying to the end user and defining available services. The simplest interactive ontological document has a form (25). Main element of interactive document is a natural system SN [19], which provides interactive access to information, contained in ontology O .

$$\langle O, SN \rangle \quad (25)$$

A natural system SN can be represented as a function (26) [19].

$$\overset{\sigma}{y} = \overset{\sigma}{f}(\overset{\sigma}{x}^1 \dots \overset{\sigma}{x}^n) \quad (26)$$

where, $\overset{\sigma}{x}^i$ – are actions performed by a user when interacting with information, $\overset{\sigma}{y}$ – the result of the work of the system in the form of text and graphical information, $\overset{\sigma}{f}$ – the target function of the system, which can be represented as (27).

$$\overset{\sigma}{f}(\overset{\sigma}{x}^1 \dots \overset{\sigma}{x}^n) = D(Q_{n-1}(\dots Q_1(X, \overset{\sigma}{x}^1) \dots, \overset{\sigma}{x}^{n-1}), \overset{\sigma}{x}^n) \quad (27)$$

where X – the set of objects of the initial ontology O , Q_i – information processing functions, D – information display function.

Examples of information processing functions include the functions of hierarchical (28) and attribute (29) filtering. [1, 2].

$$Q_h(X, x^*) = \{\bar{x} \in X | \bar{x} R x^*\} \quad (28)$$

where X – the set of ontology objects, or a certain subset of them, x^* – the object that is being filtered, \bar{R} – the specific relation between objects.

$$Q_a(X, A) = \{\bar{x} \in X | A \cap A_{\bar{x}} = A\} \quad (29)$$

where A – attributes that are filtered, A_x – attributes of an object x .

These functions allow solving a wide range of problems in displaying a variety of information, including information obtained from spatially distributed sources, which can have a different format and structure. However, when solving specific problems, such as building an oceanographic databank, these functions may be insufficient. In particular, most of the data intended for placement in the bank has a geospatial component, of which the GIS application is a natural way of representing [1, 3, 13, 20–23].

In order to ensure the maximal efficiency of processing of the information represented as the interactive document, its natural system should take into account the characteristics of the problem being solved. Moreover, in more complex subject areas (which include oceanography) various types of data can exist, which radically differ in structure and require special approaches for processing and displaying. This requires the availability of effective means of configuring natural systems, which will allow changing their composition and structure with minimal efforts. Such changes may be intended for:

- Adding specialized information display functions (in particular, for oceanographic data, which should be displayed as GIS application);
- Adding specialized information processing functions (selection of measurements in a specific date range, selection of measurements in a specific geographical region, etc.);
- Adding specialized data preprocessing functions (clustering objects on a map, interpolating measurements, etc.);

Specialized ontologies (control ontologies) can be used as a mechanism for defining the configuration of the natural system.

4.1 Ontological Presentation Templates

An ontological presentation template is a kind of control ontology designed to define additional modules of a natural system. These modules can be included in the structure of the system, changing its behaviour in accordance with the requirements of the task.

Using an ontological template within an interactive document will turn it from a pair (25) to a triple (30).

$$\langle O, O_D, SN \rangle \quad (30)$$

where O_D – ontological presentation template.

The generalized information model of the natural system, which supports ontological presentation templates, should have the form (31)

$$\alpha_{SN} = \sum_{i=0}^n \alpha_{SN}^i \cup G_T(O_D) \quad (31)$$

where α_{SN}^i – standard modules that provide basic data processing and display functions, G_T – the transformation of interpretation of an ontological presentation template.

The basic standard module of such a natural system is the system controller α_0 , which controls interactions between other modules as they perform a target function

of the system (27). The system controller provides the transformation of integration of functions of individual modules (32).

$$G_C : \bigcup_{i=0}^n S_i \cup S_T(O_D) \rightarrow \overset{\sigma}{f} \quad (32)$$

where G_C – the transformation of integration, S_i – functions of standard modules α_{SN}^i , $S_T(O_D)$ – functions, obtained from interpreting an ontological representation template. Interpreting is divided into two parallel processes – the creation of data processing functions and data display functions (33).

$$S_T(O_D) = \sum_{x \in X_D} G_Q(x) \cup \sum_{x \in X_D} G_D(x) \quad (33)$$

where X_D – a set of objects belonging to an ontology O_D , G_Q and G_D – interpretation transformations intended for the creation of data processing and presentation functions, respectively.

The use of transformations in the form (33) means that the natural system will contain many information presentations functions, which in the general case cannot be used simultaneously. Therefore, the structure of the target function (27) enables the user to select the display function that will work at the moment. For this, an auxiliary switch function (34) should be provided by the system controller α_0 .

$$D_0(X, \overset{\sigma}{x}) = \begin{cases} D_1(X), \overset{\sigma}{x} = \overset{\sigma}{x}^1 \\ D_2(X), \overset{\sigma}{x} = \overset{\sigma}{x}^2 \\ \dots \\ D_n(X), \overset{\sigma}{x} = \overset{\sigma}{x}^n \end{cases} \quad (34)$$

where X – a certain set of objects (usually the result of the processing of information processing functions), $\overset{\sigma}{x}$ – the user command that determines the choice of display function, D_i – display functions provided by system modules, $\overset{\sigma}{x}^i$ – markers of display functions.

Usage of ontological presentation templates requires an extension of the system for generating interactive documents. It must include subsystems that will provide G_Q and G_D . Their use may require additional computational resources, especially during the development period, when constant changes to ontological presentation templates are expected. Also, their use significantly increases the requirements for protection against unauthorized access to ontologies O_D . Therefore, the use of ontological presentation templates after ending of the development period may be impractical – it will be more efficient to create an interactive document with the form (25) so that its natural system already includes all the ontologically described modules as standard modules α_i .

4.2 Ontological Descriptors for Software Integration

During the creation of the oceanographic databank, the main type of data being handled are various measurements. The number of such measurements can be very significant,

but at the same time, they may be divided into subsets that are characterized by relative uniformity of data structure inside them. It is not practical to store such information in the form of an ontology due to the insufficient efficiency of the standard ontology storage (file system) when working with structured data. However, when working with such a complex subject area as oceanography, flexible mechanisms for describing the complex relationships that the ontology provides are necessary. Combined information storage allows for solving this problem. It consists of a certain specialized information system (usually a thematic database) and an ontology containing:

- The list and relationships of data types used in the framework of the information document;
- Metadata for accessing information systems used to store the data, displayed as part of an interactive document – addresses of access points, data formats, authorization data, etc.;
- Information on the composition and structure of the data displayed as part of an interactive document – a list of fields intended for display, their types and valid values;
- Information on the composition and structure of the data on the basis of which processing will be carried out – a list of fields on the basis of which filters will be formed, their possible values (for building drop-down lists), marks on geospatial information (for working with specialized functions of a GIS application);
- Information on the preliminary processing of data received through information systems – formatting and translations of field names and values;
- Another important for the correct operation of the interactive document data (for example, the location of static resources, such as source files).

The overall structure of an interactive document with a combined data storage is similar to (25), with the ontological descriptor of software integration O_P used as an ontology O . However, the natural system of such a document differs in that most of the information processing is done by underlying data storage. Therefore, the objective function of the natural system (27) is transformed into (35).

$$\tilde{f}(\overset{\sigma}{x}^1 \dots \overset{\sigma}{x}^n) = D(Q_P(x, \overset{\sigma}{x}^1, \dots, \overset{\sigma}{x}^{n-1}), \overset{\sigma}{x}^n) \quad (35)$$

where x – an object belonging to ontology O_P , which describes integration with a particular information system, Q_P – a software integration function that performs the following processes in succession:

- Forming a request to the information system based on user commands $\overset{\sigma}{x}_i$;
- Sending request and receiving information from underlying storage;
- Validating the information received, identifying errors and transmitting them to the system controller for displaying to the user;
- Pre-processing of the received information – formatting, translation etc.
- Receiving and transmitting dynamic metadata to the controller, such as the total number of results matching the query.

An interactive document with support for combined data storages should have a specialized controller that dynamically recognizes the type of the current ontology and,

depending on it, forms the target function of a natural system of the form (27) or (35). In this case, all aspects related to the display function D will remain unchanged as a result of the software integration function Q_P should not differ in structure from the result of standard information processing functions Q and should be a set of objects of dynamically formed ontology O_P^* .

In the general case, such a dynamically formed ontology will not contain ontological relations (36). This makes some display functions unsuitable for working with it.

$$R_P^* = \emptyset \quad (36)$$

In general using ontological descriptors of program integration O_P also requires using ontological presentation templates O_D and therefore an interactive document using combined information storage will have the form (37).

$$\langle O_P, O_D, SN \rangle \quad (37)$$

Using ontological integration descriptors can significantly increase the efficiency of the natural system by optimizing the information transfer process: only the information that the expert needs at a certain time is transmitted, and the pre-processing mechanism further increases the efficiency of transfer.

This mechanism provides the administrator with a standardized mechanism for controlling the system, since the ontological descriptor can be edited using standard tools like a normal ontology.

5 Oceanographic Data Processing

The main problem in the task of creating the oceanographic databank is fragmentation and incompleteness of the original data. Available data differ in structure (files of different types, including weakly structured and unstructured documents) and physical location (data may belong to different organizations, including international). In addition, the data often have incomplete information – they may not provide insights on the platform from which the measurements were taken, the measurement tool, etc. However, as practice has shown, most data have basic information, namely the type of measurements and geographical coordinates of the measuring point, which allows to normalize the data to some extent and bring them to the same structure.

As part of the first iteration of the databank creation, the following types of data were processed:

- Scientific reports concerning various measurements during expeditions (natural-language texts);
- Results of salinity and seawater temperature measurements, automatically collected from various sources (mainly ships and buoys) in various specialized formats;
- Results of hydroacoustic studies of major rivers of Ukraine and some coastal areas in specialized formats;
- Flow measurements from buoys in specialized formats.

The available data were conditionally divided into the following types:

- Point measurements – characterized by a relatively small amount of measurement at a single point on a stationary platform. Such measurements include most measurements using ships (except hydroacoustics);
- Continuous measurements – measurements taken continuously over a certain (relatively small) period of time (mainly hydroacoustic measurements);
- Long-term measurements – measurements taken over a considerable (several months) period, usually with a fixed platform (mostly buoys).

On the basis of data analysis and classification, the structure of an individual element of the databank was proposed (38).

$$d = \langle g, t, p, c, \tilde{g}, I, M \rangle \quad (38)$$

where g – geographical coordinates of the measuring point, t – measurement time information, p – identifier of the platform from which the measurements were taken, c – identifier of a series of measurements (for example, a cruise of a ship), \tilde{g} – geographical data on the further route of the platform (for mobile platforms, including ships), I – a set of related static resources (photos, source files, graphs, etc.), M – measurement data in form of a matrix.

For the main categories of input files, different approaches were taken and different processing procedures created.

5.1 Scientific Reports

The main type of documents used as input are scientific reports on oceanographic expeditions carried out by various agencies of the National Academy of Sciences of Ukraine. These reports are written in Ukrainian or Russian and contain basic information about a particular study, which in the general case is easy enough to process and bring to the structure (38).

To process such documents, the recursive reduction method is used without additional modifications. Processing requires the usage of dictionaries of attribute names, based on which information processing rules are created. The dictionaries include different variants of attribute names, which can be found in various reports (e.g. variant of a name in different languages).

Sometimes large amounts of data in scientific reports are presented in form of tables, which require some preprocessing to be efficiently used with the method of recursive reduction. Preprocessing includes restructuring such tables to form sequences of related data, required for applying recursive reduction rules.

5.2 Measurement Result Files

A feature of scientific reports is that the information they contain is optimized for human perception. As the person's ability to perceive textual information is rather limited, such optimization may require some simplification – omitting minor indicators, reducing

the number of displayed measurements, etc. This can cause problems, especially if the measurement results are collected automatically and their volumes do not allow them to be fully included in the reports. In most cases, it is more efficient to process the original files with the measurement results, ignoring the already processed data from the reports.

Many of the measurement result files represent simple tables, containing lists of measurement points. They are processed with recursive reduction method to retrieve metadata, contained in natural-language parts (commentaries) and to normalize measurement names according to given dictionaries.

Other measurement files are given in specialized formats, which require various pre-processing to be used with the method of recursive reduction. Those preprocessing may include using external libraries to access data from specialized binary files, transforming non-standard values types (like “days from a certain timestamp”) to a standard form etc.

6 Oceanographic Databank Software System

The oceanographic databank software system is designed to work in a web-based network environment. The software system can work both in local area network and on the Internet, in particular, using cloud computing [24, 25]. The system is presented as an interactive document, designed to display complex sets of data, with additional ontology-defined modules for working with data that require special means of displaying.

The interactive document uses the combined storage scheme. All measurements are stored in the internal database according to the scheme (38) and metadata stored in the ontological descriptor of software integration. The ontological descriptor contains static data that describes the existing dataset. Such data include:

- Database access credentials – used by an interactive document to obtain the necessary information;
- Filter configuration – specifies a list of fields in the structure (38) that are used as filters and are displayed in the corresponding area of the interactive document. This configuration also specifies the type of field (date, coordinates, full-text search, etc.), and for fields that are configured as drop-down lists, the values of these fields are also possible;
- Dynamic language variables, i.e. field names and field values that are specific to the current dataset.

The main purpose of ontological descriptors is to provide a flexible mechanism for configuring the process of using external data sources within the system. An additional task of the descriptors is an optimization of the data transfer process between an interactive document and an external data source (database server):

- Storing access credentials in the ontology allows a quick change of data sources (for example, when changing the network address of the database server), as well as enables usage of various external data sources simultaneously (adding them to single ontological descriptor) or separately (creating different ontological descriptors for different data sources);

- Filter configuration allows quick change of fields and field values that are available for filtering. This allows specifying fields that are excluded from filtering (used for display only). For example, in the general case there is no meaning to filter by the path of the platform \tilde{g} , so displaying corresponding field in the filter is impractical;
- The filter configuration serves as a system cache that stores the possible field values. Determining the possible values of the fields can be made during the creation of the interactive document, but it is a rather resource-intensive process, which is inappropriate to repeat constantly. An alternative to using a filter configuration within an ontology is to create a server-side caching system, however, this will require the transfer of this information from the server, which may slow the operation of the interactive document and cause additional load on the server;
- The configuration of language variables allows using shortened attribute names, more efficient for data transfer. Such names will then be translated in run-time, which also allows creating of multi-language interfaces.

6.1 Ontology-Based User Interface

The ontology-based user interface [21–23] provides the means to access available oceanographical data. A general view of this interface is shown in Fig. 1.

For displaying measurement data, GIS application is used. In addition, the auxiliary view mode is provided, which allows to displaying measurement data as a table.

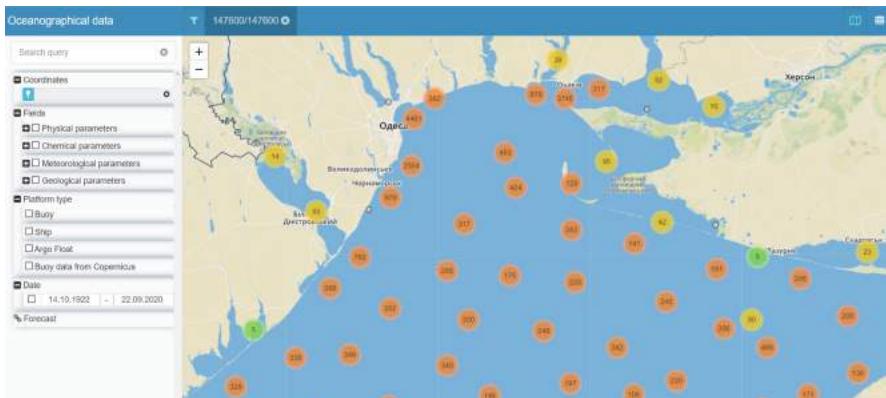


Fig. 1. The interface of the oceanographical databank

The GIS application view mode displays individual points at which measurements were made in the form of markers on the map. For convenience, large groups of markers are automatically clustered on the map. This view mode is suitable for displaying measurements on stationary platforms, like buoys or anchored ships.

For continuous measurements, this is not enough, as the trajectory of the platform movement is valuable data. If this data is available, it is stored in a part \tilde{g} from the structure (38). These data are displayed on the map in the form of thick lines, originated from a

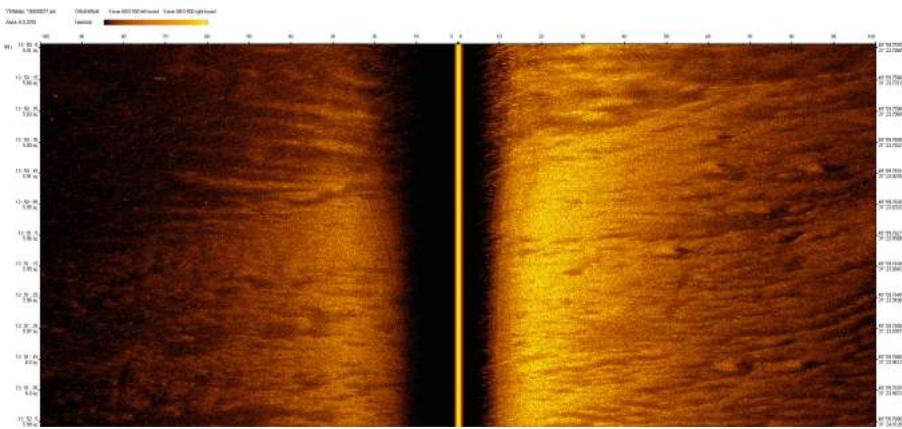


Fig. 2. Hydroacoustics measurement result

point, representing measurement. When a large amount of measurements is available, trajectories tend to intersect very often, so filtering should be used in such cases.

Some measurement types are difficult to display with standard means, so they have to be preprocessed. One such case is hydroacoustics measurements, which are processed and displayed as a set of raster images (Fig. 2).

Auxiliary table view mode (Fig. 3) allows displaying significantly fewer data at the same time then GIS application, but it provides a more detailed representation of information. Table view mode can be used to view large amounts metadata of available in the database measurements and to access measurement results.

Банк океанографічних даних									
Search query									
Coordinates									
Fields									
Physical parameters									
Chemical parameters									
Meteorological parameters									
Geological parameters									
Platform type									
Buoy									
Ship									
Argo Float									
Buoy data from Copernicus									
Date									
14.10.1922									
22.09.2020									
% Пропис									
200	200	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5994	30.9222	Bashev_04.2017.htm[2].htm	Table: 17 items(s)
201	201	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5972	31.0167	Bashev_04.2017.htm[2].htm	Table: 13 items(s)
202	202	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5952	31.0157	Bashev_04.2017.htm[2].htm	Table: 13 items(s)
203	203	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5932	31.0147	Bashev_04.2017.htm[2].htm	Table: 13 items(s)
204	204	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5984	31.0230	Bashev_04.2017.htm[2].htm	Table: 10 items(s)
205	205	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5955	31.0147	Bashev_04.2017.htm[2].htm	Table: 22 items(s)
206	206	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.5922	30.9583	Bashev_04.2017.htm[2].htm	Table: 40 items(s)
207	207	Ship	Bashev	Bashev_04.2017.htm[2]	05.04.2016	46.4917	30.975	Bashev_04.2017.htm[2].htm	Table: 37 items(s)
208	208	Ship	Bashev	Bashev_04.2017.htm	04.04.2017	46.3278	30.822	Bashev_04.2017.htm.htm	Table: 18 items(s)
210	210	Ship	Bashev	Bashev_04.2017.htm	04.04.2017	46.2972	30.8028	Bashev_04.2017.htm.htm	Table: 24 items(s)
211	211	Ship	Bashev	Bashev_04.2017.htm	04.04.2017	46.2942	30.7994	Bashev_04.2017.htm.htm	Table: 20 items(s)
212	212	Ship	Bashev	Bashev_04.2017.htm	04.04.2017	46.3167	30.7991	Bashev_04.2017.htm.htm	Table: 20 items(s)
213	213	Ship	Bashev	Bashev_04.2017.htm	04.04.2017	46.3167	30.7	Bashev_04.2017.htm.htm	Table: 14 items(s)
214	214	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5931	30.9417	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
215	215	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5928	30.9333	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
216	216	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5967	30.9222	BGKKapitanBashev_04.2017.o	Table: 3 items(s)
217	217	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.6964	30.9222	BGKKapitanBashev_04.2017.o	Table: 2 items(s)
218	218	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5972	31.0167	BGKKapitanBashev_04.2017.o	Table: 2 items(s)
219	219	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5978	31.033	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
220	220	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5978	31.0417	BGKKapitanBashev_04.2017.o	Table: 3 items(s)
221	221	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5978	30.9529	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
222	222	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.5978	30.975	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
223	223	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	05.04.2017	46.3167	30.8418	BGKKapitanBashev_04.2017.o	Table: 5 items(s)
224	224	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	04.04.2016	46.2973	30.8928	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
225	225	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	04.04.2016	46.3028	30.7998	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
226	226	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	04.04.2016	46.3141	30.7301	BGKKapitanBashev_04.2017.o	Table: 4 items(s)
227	227	Ship	BGKKapitanBashev	BGKKapitanBashev_04.2017.o	04.04.2016	46.3167	30.7095	BGKKapitanBashev_04.2017.o	Table: 3 items(s)
228	228	Ship	BGKKapitanBashev	BGKKapitanBashev_05.2017.o	03.05.2017	46.0778	30.4981	BGKKapitanBashev_05.2017.o	Table: 1 items(s)
229	229	Ship	BGKKapitanBashev	BGKKapitanBashev_05.2017.o	02.05.2017	46.0444	30.5472	BGKKapitanBashev_05.2017.o	Table: 3 items(s)

Fig. 3. Table view mode

The following fields corresponding to the elements of the structure (38) are available in the table mode:

- Platform type p ;
- Geographical coordinates g ;
- Cruise and ship identifiers (if available), which identify the measurement series c ;
- Date of measurement t ;
- The source file, which contains measurements, and related images are displayed as the part of collection of related documents I ;
- The measurement matrix M ;

The measurement matrix M often contains large amounts of measurements (tens or hundreds, and in some cases, thousands), so displaying it in a table alongside metadata is impractical. Therefore, measurement results are displayed on demand in a specialized interface. The main table contains only a mark indicating the number of measurements in the matrix.

Filtering is performed by using a specialized control block on the left side of the interactive document interface. The filter is fully configurable using the ontological descriptor of software integration, and contains the following fields:

- Coordinates – a specialized field that allows selecting a working area on the map. Only measurements located in the selected area are then displayed, both on the map and in the table view. Changing this field will automatically activate the GIS application view mode;
- Platform and parameter list – work as sets of switches that allow selection of measurements according to the selected properties;
- Date – allows selecting the date range in which displayed measurement should be made.

7 Conclusions

Currently, Ukraine has lost its accumulated database in the field of oceanography, created during the existence of the state. Oceanographic information flows, the structure of circulation and management of them was significantly disrupted. Largely, Ukraine's participation in international cooperation in the exchange of information and in international projects in the field of oceanography has ceased.

Now available in Ukraine, the measurement data of the oceanographic parameters of the Azov-Black Sea basin and the World Ocean as a whole are again scattered in disparate archives and databanks as at the beginning of the formation of the state, and some are completely lost. The analysis of the state of affairs in this area shows that there is currently no program to create a national information system for collecting, storing, analyzing and exchanging oceanographic measurement data and oceanographic knowledge. Methods, software and technological tools for analysis, special preparation and use of oceanographic information in decision-making systems in the interests of the state, science, defense and national economy of Ukraine have to be created.

The developed information system can be used as a basis for the creation of a databank, which will ensure the preservation and protection of data at the state level and can allow the country to resume its participation in international cooperation in this field.

At the same time, together with the creation of the technical and technological parts of the databank, it is necessary to implement the legal component of such system at the domestic and international level, which will allow such databank to fulfill its tasks with maximum efficiency.

References

1. Prykhodniuk, V.: Technological means of transdisciplinary representation of geospatial information. Institute of Telecommunications and Global Information Space, Kyiv (2017)
2. Stryzhak, O., Prychodniuk, V., Podlipaiev, V.: Model of transdisciplinary representation of GEOSPATIAL information. In: Ilchenko, M., Uryvsky, L., Globa, L. (eds.) UKRMICO 2018. LNEE, vol. 560, pp. 34–75. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16770-7_3
3. Prihodniuk, V., Stryzhak, O.: Ontological GIS, as a means of organizing geospatial information. Sci. Technol. Air Force Ukraine **2**(27), 167–174 (2017)
4. Guide to Operational Procedures for the collection and exchange of JCOMM Oceanographic Data. IOC Manuals and Guides No. 3 (3rd Rev. Ed.). UNESCO (1999)
5. Golodov, M., et al.: Hydrophysical research of marine and riverine environments. Geofizicheskiy Zhurnal **6**(41), 111–127 (2019)
6. Prykhodniuk, V.: Taxonomy of natural-language texts. Inf. Models Anal. **5**(3), 270–284 (2016)
7. Velychko, V., Popova, M., Prykhodniuk, V., Stryzhak, O.: TODOS - IT-platform for the formation of transdisciplinary informational environments. Syst. Arms Mil. Equipment **1**(49), 10–19 (2017)
8. Palagin, A., Kryvyyi, S., Petrenko, N.: Knowledge-oriented information systems with the processing of natural language objects: the basis of ethodology, architectural and structural organization. Control Syst. Comput. **3**, 42–55 (2009)
9. Palagin, A., Kryvyyi, S., Petrenko, N.: Ontological Methods and Means of Processing Subject Knowledge: Monograph. Volodymyr Dahl East Ukrainian National University, Luhansk (2012)
10. Stryzhak, O.: Transdisciplinary integration of information resources. Institute of Telecommunications and Global Information Space, Kyiv (2014)
11. Gladun, V., Velychko, V.: Contemplation of natural-linguistic texts. In: XIth International Conference “Knowledge–Dialogue–Solution” Proceedings, Sofia (Bulgaria), pp. 344–347. ITHEA (2005)
12. Velychko, V., Voloshin, P., Svitla, C.: Automated creation of the thesaurus of the terms of the subject domain for local search engines. In: XVth International Conference “Knowledge–Dialogue–Solution” Proceedings, Sofia (Bulgaria), pp. 24–31. ITHEA (2009)
13. Velychko, V., Prykhodinuk, V., Stryzhak, A., Markov, K., Ivanova, K., Karastanov, S.: Construction of taxonomy of documents for formation of hierarchical layers in geo-information systems. Inf. Content Process. **2**(2), 181–199 (2015)
14. Barendregt, X.: Lambda-Calculus. Its Syntax and Semantics. World, Moscow (1985)
15. Prykhodinuk, V., Stryzhak, O.: Multiple characteristics of ontological systems. Math. Model. Econ. **1–2**(8), 47–61 (2017)
16. Velychko, V., Prykhodniuk, V.: Method of automated allocation of relations between terms from natural language texts of technical subjects. In: XVth International Conference “Knowledge–Dialogue–Solution” Proceedings, Sofia (Bulgaria), pp. 27–28. ITHEA (2014)

17. Prykhodniuk, V.V., Stryzhak, O.Y., Haiko, S.I., Chepkov, R.I.: Information-analytical complex of support of transdisciplinary researches processes. *Environ. Saf. Nat. Resour.* **4**(28), 103–119 (2018)
18. Stryzhak, O., Potapov, H., Prykhodniuk, V., Chepkov, R.: Evolution of management – from situational to transdisciplinary. *Environ. Saf. Nat. Resour.* **2**(30), 91–112 (2019)
19. Malyshevsky, A.: Qualitative Models in the Theory of Complex Systems. Science, Fizmatlit, Moscow (1998)
20. Tsvetkov, V.: Geoinformation Systems and Technologies. Financial Statistics, Moscow (1998)
21. Popova, M.: Ontology of interaction in the environment of the geographic information system. Institute of Telecommunications and Global Information Space, Kyiv (2014)
22. Popova, M., Stryzhak, O.: Ontological interface as a means of representing information resources in GIS-environments. *Scientific notes of Taurida National V.I. Vernadsky University* **26**(65), 127–135 (2013)
23. Popova, M.: A model of the ontological interface of aggregation of information resources and means of GIS. *Inf. Technol. Knowl.* **7**(4), 362–370 (2013)
24. Globa, L., Sulima, S., Skulysh, M., Dovgyi, S., Stryzhak, O.: Architecture and operation algorithms of mobile core network with virtualization. In: *Mobile Computing*, pp. 427–434. IntechOpen, Rijeka (2019)
25. Globa, L., Kovalskyi, M., Stryzhak, O.: Increasing web services discovery relevancy in the multi-ontological environment. In: Wiliński, A., El Fray, I., Pejaś, J. (eds.) *Soft Computing in Computer and Information Science*. AISC, vol. 342, pp. 335–344. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-15147-2_28



Analyzing Societal Bias of California Police Stops Through Lens of Data Science

Sukanya Manna^(✉) and Sara Bunyard

Santa Clara University, Santa Clara, CA 95053, USA
smanna@scu.edu, sbunyard@alumni.scu.edu

Abstract. The Police Department in any city uses a wide range of enforcement to ensure public safety and traffic stops are one such tool. During this process, racial disparities in policing in the United States is not very uncommon. This has created both social and ethical concerns among people. As a result researchers in different domains have taken interest in connecting different pieces of information and finding a possible solution to these issues. In this paper, we have addressed the societal bias through the lens of data science focusing on how police stops are made and which groups of people are targeted most often. The analysis is done on major cities of California focusing on the different factors that might create discrimination in police stops. We also used two popularly known statistical analysis *benchmark test* and *veil of darkness* to look into racial profiling. Based on our analysis, it was clear that the Black drivers were stopped between 2.5 to 3.7 times more than white drivers and Hispanic drivers were stopped between 0.968 to 1.39 times more than white drivers (between 2014 and 2017 respectively).

Keywords: Bias · Racial discrimination · California police stops · Benchmark test · Veil of darkness test

1 Introduction

Background and Motivation: The Police departments use a wide range of enforcement tools to ensure public safety. Traffic stops are one such tool. These interactions typically involve an officer pulling over a motorist, issuing a warning or citation, and more rarely conducting a search for contraband or making a custodial arrest [4]. The prevalence and nature of traffic stops vary widely across American cities, but they are generally the most common way police departments initiate contact with the public [5].

There are several studies which have looked into racial profiling and used different statistical methods to quantify discrimination. The key problem in testing for racial profiling in traffic stops is estimating the risk set, or “benchmark”, against which to compare the race distribution of stopped drivers [8]. To date, the two most common approaches have been to use residential population data

or to conduct traffic surveys in which observers tally the race distribution of drivers at a certain location. It is widely recognized that residential population data provide poor estimates of the population at risk of a traffic stop; at the same time, traffic surveys have limitations and are more costly to carry out, so the *veil of darkness* test is applied instead by Grogger and Ridgeway [8]. There are other well-known statistical tests for measuring discrimination; some of them are *Benchmark test*, *Outcome test*, *Threshold test* and different variants of them.

According to [13], the authors discussed possible racial disparities as black and Hispanic drivers were often stopped and searched on the basis of less evidence than whites. To reach this conclusion, search rates were examined. In nearly every jurisdiction, it was found that stopped black and Hispanic drivers were searched more often than whites, about twice as often on average. But such statistics alone are not clear evidence of bias. If minorities also happen to carry contraband at higher rates (a hypothetical possibility, not a fact), these elevated search rates may simply reflect routine police work. This raises concerns among local community groups that traffic stop practices disproportionately and impact certain races over others.

There have been initiatives to analyze the underlying causes of this. For example [12], in May 2014, the City of Oakland and the Oakland Police Department (OPD) partnered with the team of Stanford social psychologists to collect and analyze data on OPD officers' self-initiated stops. The task of the research group was to analyze the reports that OPD officers completed after every stop they initiated between April 1, 2013 and April 30, 2014. The analyses revealed that OPD officers disproportionately stopped, searched, handcuffed, and arrested African Americans, relative to other racial groups.

Our main motivation is to find if a similar scenario prevails at the same depth in California, as the state itself is considered to be very diverse and multicultural. Some of the earlier research focused on either USA as a whole or other states like Texas [11], New York [13] North Carolina [2], on racial profiling.

Contributions: In this paper, we present three-fold analysis: First, we aim to quantify racial disparities in California's current traffic stop practices using two main statistical analysis, *Benchmark test* and *Veil of Darkness*. Second, we visualized for both pedestrian and vehicular level stops at prime California cities. Third, we focused on stops in 2014 and 2017 to compare and contrast if there was any change in the pattern of policing.

Organization: We have explained these two goals in the remainder of this paper. Section 2, presents the related work. Section 3 presents the experimental setup. Section 4 illustrates our findings, and finally Sect. 5 concludes the paper.

2 Related Work

There has been several studies on discrimination seen in the behavior of police stops towards drivers or pedestrians based on different factors like gender or race or ethnicity. It is seen from several instances that police officers speak

significantly less respectfully to black than to white community members in everyday traffic stops, even after controlling for officer race, infraction severity, stop location, and stop outcome [14]. Researchers from various domains have gathered and analyzed data about these stops [2,7,9].

There are well-known statistical tests for measuring discrimination; some of them are *Benchmark test*, *Outcome test*, *Threshold test* and different variants of them. We have only focused on the ones closest to the analysis we have done in this paper.

In the first test, termed benchmarking, compares the rate at which whites and minorities are treated favorably. However, there is one limitation of benchmarking referred to in the literature as the qualified pool or denominator problem [1], and is a specific instance of omitted variable bias. For example, while looking at the police stops per capita, the driving population might be different from the census (or residential) population.

Addressing this shortcoming of benchmarking, Becker [3] proposed the *outcome test*, which is based not on the rate at which decisions are made, but on the success rate of those decisions. These theories were based off granting credit loans. Though originally proposed in the context of lending decisions, outcome tests have gained popularity in a variety of domains, particularly policing [1,6,7,10]. For example, when assessing bias in traffic stops, one can compare the rates at which searches of white and minority drivers turn up contraband. If searches of minorities yield contraband less often than searches of whites, it suggests that the bar for searching minorities is lower, indicative of discrimination.

Besides benchmarking and outcome, *threshold tests* [13] have recently been proposed as a useful method for detecting bias in lending, hiring, and policing decisions. For example, in the case of credit extensions, these tests aim to estimate the bar for granting loans to white and minority applicants, with a higher inferred threshold for minorities indicative of discrimination. This technique, however, requires fitting a complex Bayesian latent variable model for which inference is often computationally challenging. So *Fast threshold test* was developed. It is a method for fitting threshold tests that is two orders of magnitude faster than the existing approach, reducing computation from hours to minutes. To illustrate their efficacy, these algorithms were tested against 2.7 million police stops of pedestrians in New York City.

3 Experimental Setup

Dataset: Our data comes from the Stanford Open Policing Project¹. The Stanford Open Policing Project is a partnership between the Stanford Computational Journalism Lab and the Stanford Computational Policy Lab that collected and standardized over 200 million records of traffic stop and search data from across the country.

¹ <https://openpolicing.stanford.edu/data/>.

We have used 8 out of 11 the cities of California's data which is provided in the dataset (*Bakersfield, Long Beach, Los Angeles, Oakland, San Diego, San Francisco, San Jose, Stockton*) as not every city here contained the "race" information of the people being stopped by the police. Since "race" is one of main attributes responsible to find out discrimination, we discarded *Anaheim* and *San Bernardino*. We also discarded *Santa Ana*, because there were not enough months of data overlap with the other cities. In many cases, the data we used were insufficient to assess racial disparities (for example, the race of the stopped driver was not regularly recorded, or only a non-representative subset of stops was provided). For consistency in our analysis, we further restricted stops occurring in 2014 and 2017 as many jurisdictions did not provide consistent data for other years. Our primary dataset for our analysis thus consists of approximately 1,044,030 stops.

Data Processing: We processed the data in few steps:

1. *Merging multiple files:* The different cities provided differently formatted data, and while the data from the Open Policing Project was already cleaned and formatted, some cities provided different variables than others. So we merged the files with the necessary modifications.
2. *Data cleaning:* When cleaning the data, we filled in the time and driver gender variables with 'N/A' because some of our cities did not have this data recorded. Once this was complete, we eliminated observations with missing values. We then restricted our data to the year 2014, because that was the year with the most complete data. Then, we eliminated rows from Santa Ana because it did not have complete data for 2014 and 2017. Lastly, we created hour and weekday variables that extracted the hour from the stop time and the weekday from the stop date when that information was available.

4 Analysis

We divided our analysis into two phases. Firstly, we presented different variables which might create a "biased" decision in stops. Secondly, we did statistical analysis "Benchmark test" and "Veil of darkness" to analyze racial disparities.

4.1 Analysis of Stops Through Visualizations

One of the criteria for this analysis was that the data needed to be present consistently in all the cities during a certain time span. We recorded that the years 2014 & 2017 (relatively most recent data) had maximum overlap in terms of the time, so we retained them for comparisons. This analysis included both vehicular and pedestrian stops. We looked at the outcome of the stops based on gender and race primarily. We also compared the frequency of police stops at different times of the day and different days of a week just to have a better understanding of policing.

The number of data points for the year 2014 is different from 2017 for our visualization because fewer cities had data available for 2017 (Table 4). For the 2014 visualizations, the total number of stops is 1,044,030. Distribution of stops per city is summarized in the Tables 1 and 2. We also examined some cities in 2017 to look at how the data had changed. We chose the year 2017 because it was the most recent year where the most cities had complete data available. The cities that we were able to use were Los Angeles, Oakland, San Jose, Bakersfield and Long Beach. Our dataset for 2017 contained approximately 693,980 stops.

Table 1. Distribution of stops per city in 2014

City	Stops
Bakersfield	22874
Long Beach	23355
Los Angeles	704705
Oakland	23062
San Diego	138917
San Francisco	91961
San Jose	32260
Stockton	6896

Table 2. Distribution of stops per city in 2017

City	Stops
Bakersfield	19129
Long Beach	18475
Los Angeles	598275
Oakland	30545
San Jose	27556

Per Capita Stops by City: Figures 3 and 4 illustrate per capita stops by city for the years 2014 and 2017. For this, we took our stop data and grouped it by driver race. Then, we joined it with census data from the 2010 census. To get our census data, we took the total population and multiplied it by the different demographic percentages to get the number of people of each demographic in each city. Once these two datasets were joined together, for each city we divided the number of stops of each race by the number of people of each race in that city, to get stops per capita by race. To get our confidence intervals for all our data combined, we took the unweighted average of the stop rates to get our combined per capita stops for each race. Then we used the formula:

$$CI = (\hat{p}_1 - \hat{p}_2) \pm z \sqrt{\hat{p}_1 \frac{1 - \hat{p}_1}{n_1} + \hat{p}_2 \frac{1 - \hat{p}_2}{n_2}} \quad (1)$$

where \hat{p}_1 is the proportion of stops of $race_1$, \hat{p}_2 is the proportion of stops of $race_2$, n_1 is the population sum of $race_1$, and n_2 is the population sum of $race_2$, respectively. z is the z -score at a certain confidence interval (for our case its 99% confidence interval).

Based on Eq. (1), we computed confidence intervals by comparing White and Hispanic Drivers in 2014 (99% CI [0.002, 0.003], $p < 0.01$) and 2017 (99% CI [-0.02, -0.02], $p < 0.01$). For 2014 and 2017, the results are close to zero, making it insignificant for us to present. Similarly, we computed confidence intervals for White and Black in 2014 (99% CI [-0.114, -0.111], $p < 0.01$) and 2017 (99% CI [-0.134, -0.132], $p < 0.01$) respectively.

Now, analyzing per capita stops by city, in 2014, it is seen that Black drivers are stopped 2.52 times more than white drivers and in 2017, 3.73 times more than white drivers (see Tables 3 and 4). Similarly in 2014, Hispanic drivers are stopped 0.968 as many times as white drivers and they are stopped 1.39 times more than white drivers in 2017.

Table 3. Per capita stops for each city of drivers from different races in 2014

City	Black drivers	Hispanic drivers	White drivers
Bakersfield	0.114	0.035	0.108
Long Beach	0.0976	0.0443	0.0429
Los Angeles	0.471	0.164	0.152
Oakland	0.129	0.0442	0.0331
San Diego	0.182	0.107	0.107
San Francisco	0.334	0.0992	0.103
San Jose	0.114	0.0597	0.0244
Stockton	0.0527	0.0219	0.0239

Table 4. Per capita stops for each city of drivers from different races in 2017

City	Black drivers	Hispanic drivers	White drivers
Bakersfield	0.074	0.0472	0.0595
Long Beach	0.0794	0.038	0.0302
Los Angeles	0.469	0.144	0.108
Oakland	0.2	0.0632	0.0251
San Jose	0.0852	0.0469	0.0204

Stops by Gender: Figure 1a and 1b are visualizations of stops by gender in these two years. For both cases, it is quite obvious that females are stopped

less often than men but this claim does not justify the fact that the population of females might be lesser than males or the number of females in the driving population on road might be smaller as well.

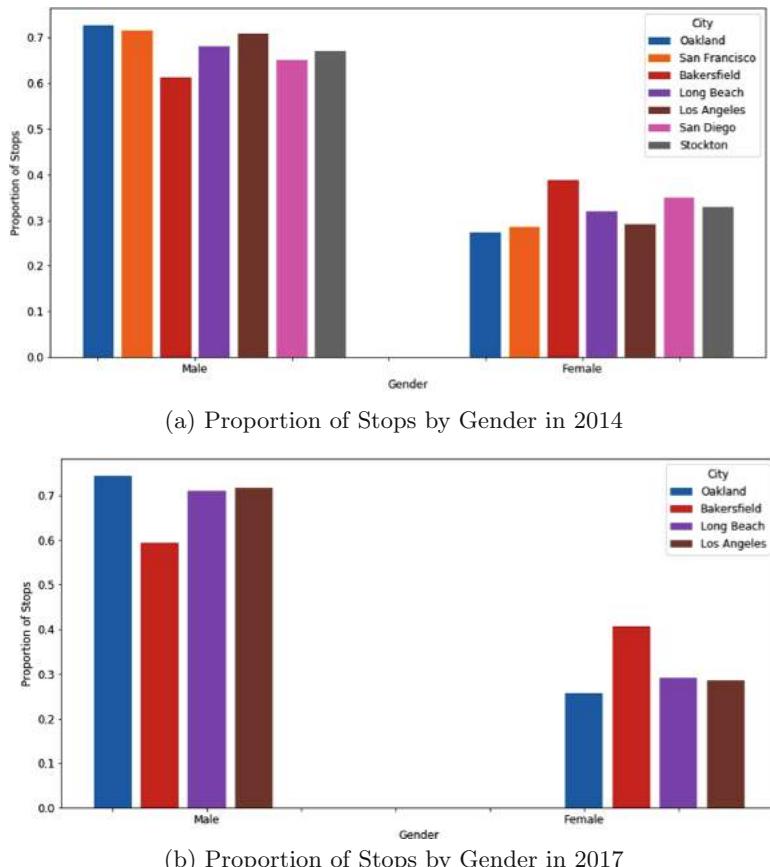


Fig. 1. Analysis of stops by gender

Stops by Race: Figure 2a and 2b are visualizations of stops by race in these two years. For computing the stops by races' visualization, total stops are 964,455. It has less data points because we discarded the points those were listed race as *other* or *unknown*. It is given in the Table 5. The figures in general reflect the fact that in cities with higher concentration of certain races, those races seem to have higher proportions of stops over others. For example, if we look at the city of San Jose, there is a higher ratio of Hispanic population existing. Now the stops are also relatively higher for Hispanic in San Jose than other cities. This does not infer anything specifically about any racial discrimination.

Table 5. Distribution of stops by races per city in 2014

City	Stops
Bakersfield	22014
Long Beach	20758
Los Angeles	655132
Oakland	22149
San Diego	130220
San Francisco	77217
San Jose	30353
Stockton	6612

Stops by Time: Figure 4a and 4b and 3a and 3b are analysis of stops by days of the week and by hours in these two years.

For each hour of the day, we calculated the percentage of stops in each city that occurred in that hour. The figure in 2014 is composed of 1,013,210 stops (Long Beach and Stockton were excluded because the time of the stops was not available.) The figure in 2017 is composed of 675,421 stops. From the figures we can see that generally stops are lower in the early morning around 5 am, and higher in the afternoon, evening, and late at night.

For day of the week, we calculated the percentage of stops in each city that occurred on that day. All of the cities had stop dates available, and the figure from 2014 is composed of 1,044,030 stops. The figure from 2017 is composed of 693,980 stops. From the figures we can see that generally more police stops are made in the middle of the week, while less are made on weekends and Mondays.

4.2 Assessing Racial Discrimination in Traffic Stop Decisions Using Statistical Analysis

We have presented two separate statistical tests to validate if there is any racial discrimination in traffic stop decisions. We have used *Benchmark Test* and *Veil of Darkness*. Each subsections describe our findings in details.

Benchmark Test: At first we had to understand the racial demographics in the California population data. We could not use the raw population number. To make it useful, computed the proportion of California residents in each demographic group per city. As an eyeball comparison led us to see that black drivers were being stopped disproportionately, relative to the city's population. Dividing the black stops per capita by the white stops per capita to be able to make a quantitative statement about how much more often black drivers are stops compared to white drivers, relative to their share of the city's population. Black drivers are stopped at a rate of 2.52 in 2014 and 3.73 in 2017 times higher than white drivers; and Hispanic drivers are stopped at a rate 0.968 as many times

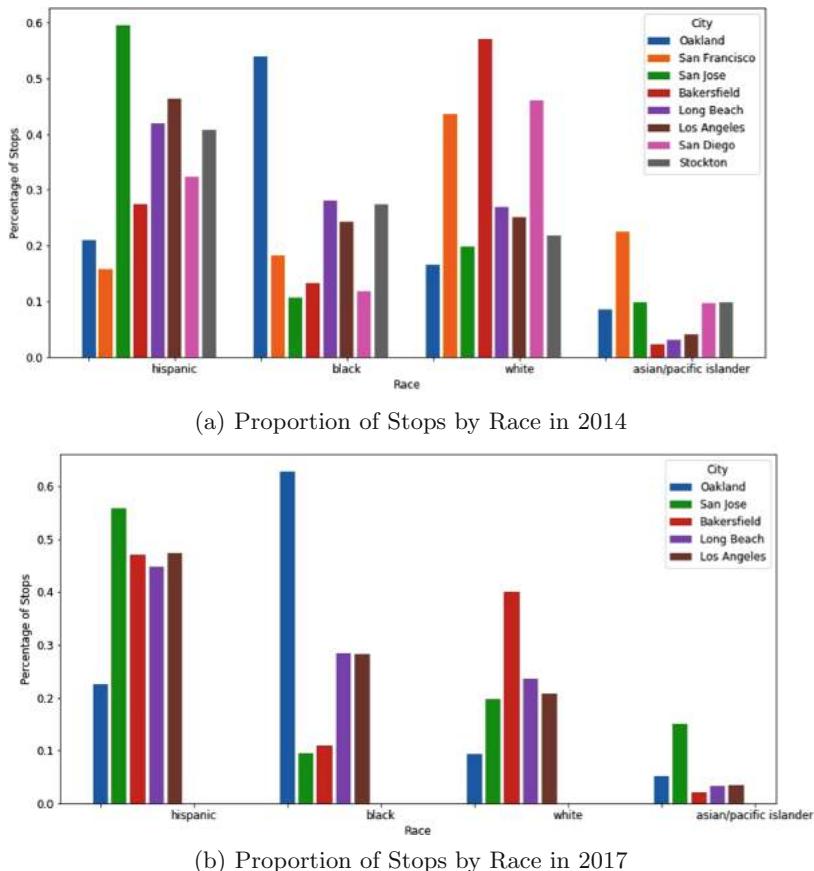


Fig. 2. Analysis of stops by race

in 2014 and 1.39 times higher in 2017 than white drivers. Since there was not enough data for other races, we decided not to compute this test for them.

Caveats about the Benchmark Test: While these baseline stats give us a sense that there are racial disparities in policing practices in California, they are not evidence of discrimination. The argument against the benchmark test is that we have not identified the correct baseline to compare to.

For the stop rate benchmark, what we really want to know is what the true distribution is for individuals breaking traffic laws or exhibiting other criminal behavior in their vehicles. If black and Hispanic drivers are disproportionately stopped relative to their rates of offending, that would be stronger evidence. Some people then proposed to use benchmarks that approximate those offending rates, like arrests, for example. However, we know arrests to themselves be racially skewed (especially for low-level drug offenses, for example), so it wouldn't give us the true offending population's racial distribution.

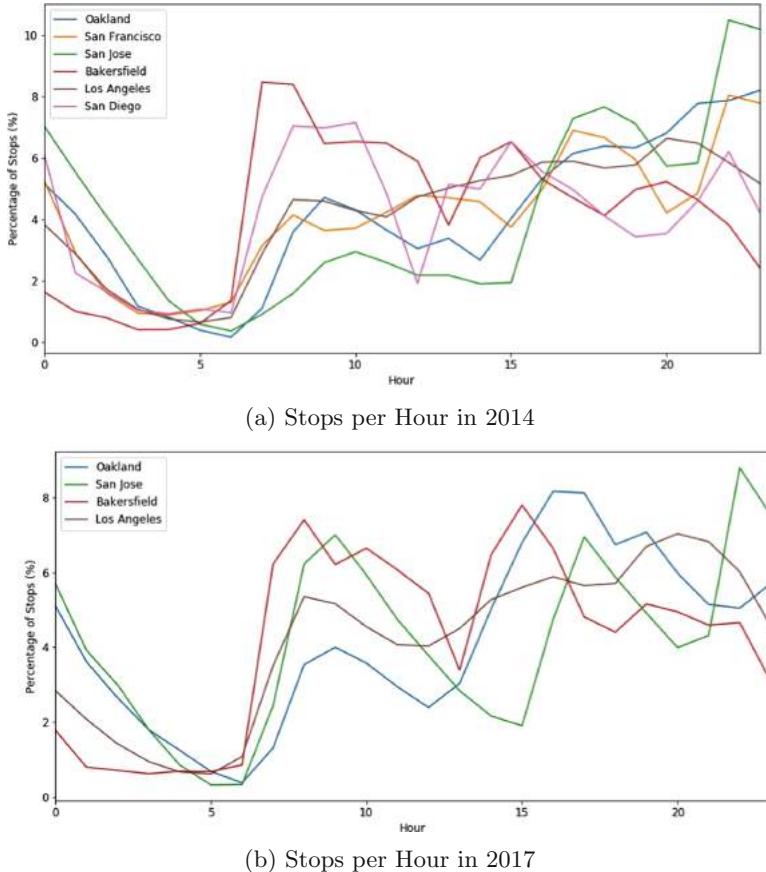
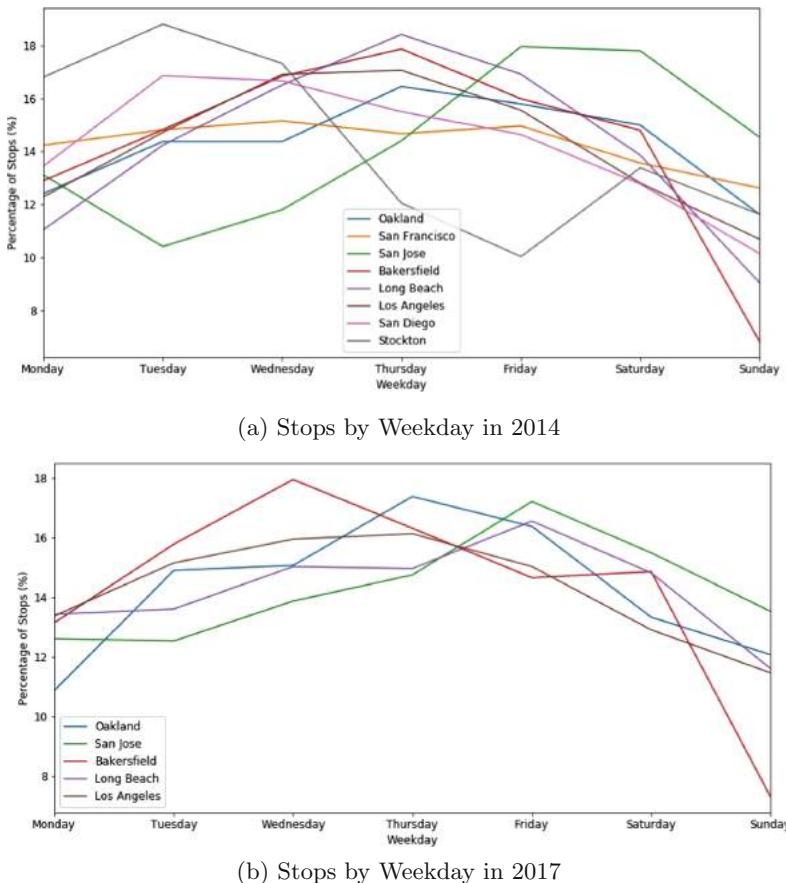


Fig. 3. Analysis of stops per hour

An even simpler critique of the population benchmark for stops per capita is that it doesn't account for possible race-specific differences in driving behavior, including amount of time spent on the road (and adherence to traffic laws, as mentioned above). If black drivers, hypothetically, spend more time on the road than white drivers, that in and of itself could explain the higher per capita stops we see for black drivers, even in the absence of discrimination.

Veil of Darkness [8] is primarily used to assess racial discrimination at the stop decision. This approach relies on the hypothesis that officers who are engaged in racial profiling are less likely to be able to identify a driver's race after dark than during daylight. Under this hypothesis, if stops made after a smaller proportion of black drivers stopped than stops made during daylight, this would be evidence of the presence of racial profiling.

The naive approach is just to compare whether daytime stops or nighttime stops have higher proportion of black drivers. This, however, is problematic.



(b) Stops by Weekday in 2017

Fig. 4. Analysis of stops per hour

More black drivers being stopped at night could be the case for a multitude of reasons: different deployment and enforcement patterns, different driving patterns, etc. All of these things correlate with clock time and would be confounding our results. So the clock time needs to be adjusted accordingly.

To adjust for clock time, we want to compare times that were light at some point during the year (i.e., occurred before dusk) but were dark at other points during the year (i.e., occurred after dusk). This is called the “inter-twilight period”: the range from the earliest time dusk occurs in the year to the latest time dusk occurs in the year.

We calculated the sunset and dusk times for California by using the library *suncalc*² which calculated sunset and dusk times on the day of each stop based on the stop date and center latitude and longitude of the stop city. Our approach is similar to [13]. For this analysis we also discarded *Long Beach* and *Stockton*

² <https://cran.r-project.org/web/packages/suncalc/suncalc.pdf>.

because the time of the stops were not available, so we did not have enough information to compute the test.

Adjustments with the Data for this Analysis: To calculate the veil of darkness analysis we chose data from the inter twilight period (when it is dark during part of the year and light during other parts of the year. We also filtered out stops that were in between sunset and dusk that could have been ambiguous.

Findings: Table 6 illustrate our result for the Veil of Darkness analysis. To estimate this, we used logistic regression, using a natural spline with 6 degrees of freedom as seen in [13] except for the fact that we only had the city level data. Equation (2) provides the logistic regression model we fitted:

$$Pr(black|t, g, p, d, s, c) = logit^{-1}(\alpha_c \times c \times d + \beta^T \times ns_6(t) + \gamma[g] + \delta[p]) \quad (2)$$

where $Pr(black|t, g, p, d, s, c)$ is the probability that a stopped driver is black at a certain time t , location g and period p (with two periods per year, corresponding to the start and end of DST), with darkness status $d \in \{0, 1\}$ indicating whether a stop occurred after dusk ($d = 1$) or before sunset ($d = 0$), and with the enforcement agency being city police department, $c \in \{0, 1\}$. In this model, $ns_6(t)$ is a natural spline over time with six degrees of freedom, $\gamma[g]$ is a fixed effect for location g and $\delta[p]$ is a fixed effect for period p . The location $g[i]$ for stop i corresponds to city, with $\gamma[g[i]]$ the corresponding coefficient. Finally, $p[i]$ captures whether stop i occurred in the spring (within a month of beginning DST) or the fall (within a month of ending DST) of each year, and $\delta[p[i]]$ is the corresponding coefficient for this period. For computational efficiency, we rounded time to the nearest 5 – min interval when fitting the model.

Table 6. Veil of darkness analysis using combined city data

	Estimate	Std. Err
Year 2014	-0.118	0.020
Year 2017	-0.061	0.027

The basic idea is that we are finding what our data implied about how darkness and clock time influence whether a stopped driver will be black. We then extract the coefficient of darkness. The fact that the coefficient is negative (Table 6) means that darkness lessens the likelihood that a stopped driver will be black, after adjusting for clock time. This matches our intuitive result from 6:30 p.m. [13]. The fact that the standard error is small means that this is a statistically significant finding.

Caveats on the Veil of Darkness Test: We argue and conform with [13] that the veil-of-darkness test, like all statistical methods, it comes with caveats. Darkness, after adjusting for time of day, is a function of the date. As such, to the extent that driver behavior changes throughout the year, and these changes are

correlated with race, the test can suggest discrimination where there is none. One way to account for seasonal effects is to consider the brief period around daylight savings. Driving behavior likely doesn't change much at, say, 6 : 00p.m. the day before daylight savings to the day after; but the sky is light on one day and dark on the next.

Another thing to note is that artificial lighting (e.g., from street lamps) can weaken the relationship between sunlight and visibility, and so the method may underestimate the extent to which stops are predicated on perceived race. Other things, like vehicle make, year, and model often correlate with race and are still visible at night, which could lead to the test under-estimating the extent of racial profiling. Similarly, the test doesn't control for stop reason, which is often correlated with both race and time of day. Finally, the test only speaks to presence of racial profiling in the intertwilight period doesn't say anything about other hours. Nevertheless, despite these shortcomings, the test provides a useful if imperfect measure of bias in stop decisions.

5 Conclusion

We analyzed 8 cities out of 11 California prime cities for this research. We have summarized our take away from this research:

1. We looked at different visualizations at different cities of California focusing on multiple attributes like race, gender, time. Due to not enough information provided in the dataset, the initial visualizations could only illustrate the scenario of police stops in a broader sense.
2. Per capita analysis projected racial discrimination by police during these stops, justifying the claims the difference in police behavior with different groups of people.
3. The benchmark test and the veil of darkness also added extra weightage to our findings on racial discrimination during police stops.

In spite of California being multicultural and diverse, we still see similar outcomes in policing.

Future Work: As a future work, we intend to extend this work in the following ways:

1. Add more robustness tests for the model by tuning different variables
2. We could look for more additional data that would allow us to compute different other parameters like search rate, arrest rate and so on, so that other popular statistical tests such as outcome and threshold tests.
3. Further research on demographics would allow us to compute more accurate per capita stops.

References

1. Ayres, I.: Outcome tests of racial disparities in police practices. *Just. Res. Policy* **4**(1–2), 131–142 (2002)
2. Baumgartner, F.R., Epp, D.A., Shoub, K., Love, B.: Targeting young men of color for search and arrest during traffic stops: evidence from North Carolina, 2002–2013. *Polit. Groups Ident.* **5**(1), 107–131 (2017)
3. Becker, G.S.: *The Economics of Discrimination*. University of Chicago Press (2010)
4. Chohlas-Wood, A., Goel, S., Shoemaker, A., Shroff, R.: An analysis of the metropolitan nashville police department's traffic stop practices. Technical report, Stanford Computational Policy Lab (2018)
5. Davis, E., Whyde, A., Langton, L.: Contacts between police and the public (2015)
6. Goel, S., Perelman, M., Shroff, R., Sklansky, D.A.: Combatting police discrimination in the age of big data. *New Crim. Law Rev.* **20**(2), 181–232 (2017)
7. Goel, S., Rao, J.M., Shroff, R., et al.: Precinct or prejudice? Understanding racial disparities in New York city's stop-and-frisk policy. *Ann. Appl. Stat.* **10**(1), 365–394 (2016)
8. Grogger, J., Ridgeway, G.: Testing for racial profiling in traffic stops from behind a veil of darkness. *J. Am. Stat. Assoc.* **101**(475), 878–887 (2006)
9. Horrace, W.C., Rohlin, S.M.: How dark is dark? Bright lights, big city, racial profiling. *Rev. Econ. Stat.* **98**(2), 226–232 (2016)
10. Knowles, J., Persico, N., Todd, P.: Racial bias in motor vehicle searches: theory and evidence. *J. Polit. Econ.* **109**(1), 203–229 (2001)
11. Liederbach, J., Trulson, C.R., Fritsch, E.J., Caeti, T.J., Taylor, R.W.: Racial profiling and the political demand for data: a pilot study designed to improve methodologies in Texas. *Crim. Just. Rev.* **32**(2), 101–120 (2007)
12. Maitreyi, A.: Data for change: a statistical analysis of police stops, searches, handcuffings, and arrests in Oakland, Calif., 2013–2014. Stanford University, SPARQ: Social Psychological Answers to Real-World Questions (2016)
13. Pierson, E., et al.: A large-scale analysis of racial disparities in police stops across the united states. *Nat. Hum. Behav.* 1–10 (2020)
14. Voigta, R., et al.: Language from police body camera footage shows racial disparities in officer respect. *PNAS* **114**(25), 6521–6526 (2017)



Automated Metadata Harmonization Using Entity Resolution and Contextual Embedding

Kunal Sawarkar and Meenakshi Kodati^(✉)

IBM, Armonk, USA
{kunal,meenakshi.kodati}@ibm.com

Abstract. Data curation process for Analytics and Data Science typically involves collecting data from large number of heterogenous and federated source systems with varied schema structures. To make these datasets interoperable, their metadata needs to be standardized. This process, also known as Metadata Harmonization, is predominantly a manual effort involving several hours of concentrated work that leads to reduced efficiency of ML-Ops lifecycle. This paper aims to demonstrate the automation of metadata harmonization using Machine Learning. It focuses on using entity resolution and contextual embedding methods to capture hidden relationships among data columns that help identify similarities in metadata, and thereby, help in automated mapping of columns to a standard schema. This study also addresses the automated derivation of the correct ontological structure for the target data model using ML. While prior competing approaches address manual metadata harmonization problem by proposing usage of semantic middleware, data dictionaries and matching rules this approach recommends novel usage of Machine Learning which improves efficacy of overall lifecycle.

Keywords: Metadata harmonization · Metadata crosswalking · Data curation · Metadata contextual embedding

1 Introduction

Large multinational organizations that gather data from disparate data sources are often faced with the challenge of standardizing the data formats to make the datasets interoperable. This involves finding similarities among data collection methodologies and creating mapping tables that meaningfully unite information from them. This method of mapping identical or equivalent metadata across varied datasets is known as *Metadata Crosswalking or Master Metadata Synchronization or Metadata Harmonization*. This includes task of metadata mapping & cleansing the metadata to given standard and linking it to a standard structure (see Fig. 1). For a large dataset which has hundreds of columns, mapping metadata to a catalog can be a big challenge.

One of the limitations of the data curation and cataloging tools available today is that they are not equipped with metadata harmonization capabilities that automate the process of crosswalking. As a result organizations rely on Data Stewards to perform this task manually, often consuming several days or weeks of concentrated effort. Data Stewards manually contrast each column and map it to a set standard schema (see Fig. 2). This being first step in curation, also becomes the bottleneck for efficiency of data science project executions. There is also a possibility that the new data proposes additions or changes to the current ontological structure that the standard schema has to follow.

The aforesaid limitations underscore the need for a data curation process that is embedded with the ability to automatically crosswalk the metadata to a standard format at the time of ingestion, and preferably enhance the metadata by deriving an ontological metadata structure that can be scaled to a wide variety of data sources automatically.

The current study addresses the above mentioned needs by (1) *Proposing a metadata harmonization method that can automatically crosswalk metadata of different data sources to a standard schema*, (2) *Autonomously deriving the standard ontological structure of metadata from a multitude of source systems using Machine Learning based Entity Resolution Methods that leverage Levenshtein distance-based mechanism to find the relevant entities by employing cost-based approaches* (3) *Using embedding methods like db2vec that involve contextual vectorization and textification to resolve metadata entities to the nearest standardized schemas*. The system worked well for inferring metadata column names in a schema as well as an ontological hierarchical structure for that schema in experimentation performed on test data.

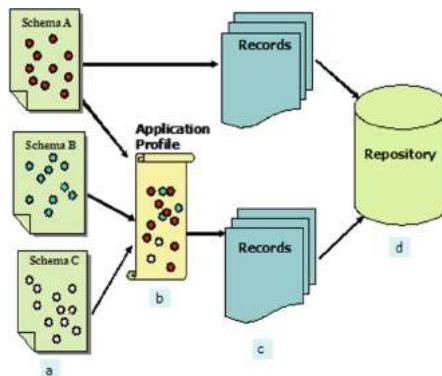


Fig. 1. Crosswallking metadata for multiple schemas

In the literature there had been limited use of ML to automate & improve ML-ops process of cataloguing. In previous attempts most of the approaches relies on a prebuilt and static data dictionary which serves as a look up function

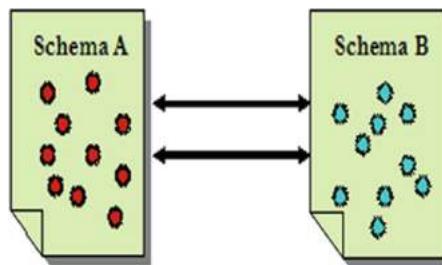


Fig. 2. Mapping ontological structure to standard metadata

for new metadata; however such methods needs apriori knowledge of the system & are not scalable to previously unknown schemas or ontological structures.

2 Related Studies

The automation of metadata harmonization as researched by Wurzer [1] can be done through the usage of a semantic middleware for data integration and content analysis that analyzes the semantic closeness of incoming data against historically developed concrete data models to find similarities in the metadata and data content. While it presents a solution to replace manual master data management, it relies on the availability of concrete data models from historically stored datasets, and on content objects that provide a generic description of the new data objects. Furthermore, it focuses on *semantic closeness* by employing concepts like regular expression matching and data closeness analysis. Therefore, it is less likely to work effectively on completely new, previously-unseen datasets. In cases where crowd sourced datasets are collected, where the likelihood of consistent data formats and rules is minimal, adopting concepts like regular expression matching and data closeness analysis may not result in accurate classification. The current paper therefore emphasizes the use of ML concepts like entity resolution and contextual embeddings for metadata harmonization. It relies on a base ontological structure to *learn* to classify and map metadata, and incrementally learns with the help of advanced AI algorithms to improve accuracy of classification. The db2vec approach [4], which formed the basis for the current study, uses contextual embedding, and can be leveraged to derive column relationships even in the absence of a base ontology by deriving and learning from canonical industry data models. Furthermore, this paper also researches automatic derivation of an ontological structure with every new column matching that happens. The strength of this approach, therefore, lies in the application of machine learning for continuous learning and improvement of the base ontological structure.

The automated derivation and refinement of the ontological structure, cited as one of the merits of the current study, addresses a key pain point for Master Data Management. The construction of ontology conventionally starts with

interviewing the business stakeholders to determine high level concepts, and is followed by manual inspection of the source systems to extract commonalities which form the foundation for a base ontology, which is then enriched with new entity types and properties. The resulting ontology is then customized for different applications and is mapped back to source system to facilitate subsequent extraction of relevant data from different sources. As the number of source systems increases, the process becomes increasingly laborious, which calls for some degree of automation. The work by Cobett et al. [5] proposes automatic ontology generation through the use of a semantic network of known concepts pertaining to a target domain of knowledge, followed by data discovery techniques that help augment the semantic network with classifications of data attribute properties. The said study too, relies on semantic closeness and availability of historic data with clearly defined technical metadata. The present research utilizes machine learning to enable classification of even those (relevant) data attributes that are completely different from other attributes in the source systems by learning the context of metadata and can even work on previously unseen metadata.

Other relevant competing approaches that aim to tackle the metadata harmonization problem include a study by Schon [2], that proposes storing specifications for different sources, using the specifications to generate rules for each source and then matching data elements of different sources to derive a quality metric that characterizes a given match. Another study by Cope [3] aims to combine business rules and data dictionary into an interlanguage document type definition to crosswalk metadata. These approaches require prior knowledge and are less likely to be scalable for new schemas.

3 System Design

This section discusses the framework design aspects for the proposed embodiments and describes their implementations in detail.

3.1 Framework Overview

The framework can be summarized as below (see Fig. 3).

1. The metadata for the crosswalk from various heterogenous sources is collected and arranged in an ontological structure. It also includes additional meta-metadata like lookup for a column attributes for a given standardization format and a data model schema structure with its own naming & descriptive structure for which crosswalk is expected to be performed. All the above metadata is stored in a data frame which can include source column names and the expected standardized schema tiers or data model tiers for the crosswalk output.
2. Based on this meta-metadata like column name and the descriptions the metadata entity are resolved using machine learning methods for the crosswalk predictions. Various embodiments can be used for entity resolution to perform schema crosswalk. Two such methods are discussed below:

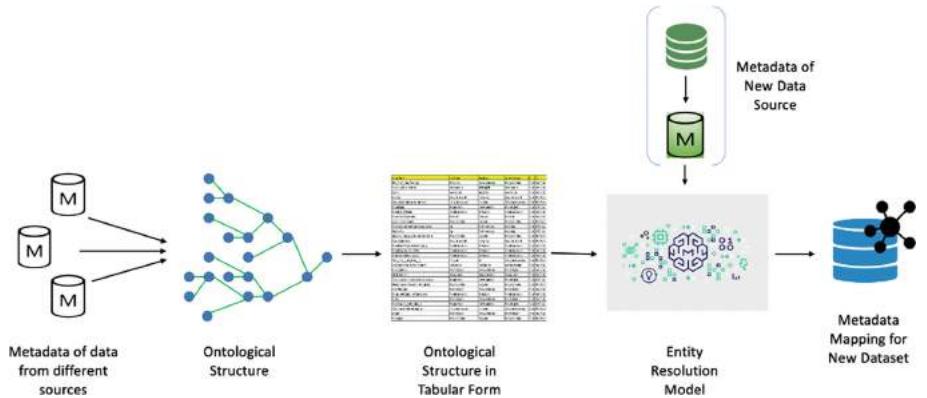


Fig. 3. Framework for the metadata harmonization using entity resolution mechanism

- (a) **Levenstein Distance Based Method:** This method, as explained in 4 uses the cost based textual distance to find entity matching on each column names and to the standardized schema of crosswalk model. For tie resolution in case of same Levenstein scores a blocked indexing method is used including a hybrid method.
- (b) **Using Contextual Embeddings:** Using *db2vec* [4] which captures semantic context for the entities and find similar entities to the crosswalk. Further we found that using method *db2vec* [4], it creates textification and then vectorization for all the entities, and matches the entities to the expected schemas by capturing column relationships.
- 3. In case there exists a ground truth for the entity matching from previous manual efforts done by data stewards then an additional text classification model can learn from previous metadata crosswalks and improve the performance. The system can also work in the absence any training data to learn explicitly context of column relationships. The solution methodology can also derive & learn metadata from canonical industry data models to specific domain (like healthcare, banking, etc.)

3.2 Implementation

The first step in the process is to obtain the ontology that can be used for the Entity Resolution model. Each metadata entity is described by the meta-metadata which can include any of the following descriptive attributes about columns viz. *a) Verbose Column Name, b) Data classification Business Terms, c) Any Textual Description of the column, d) Business Glossary, e) Data Dictionary*. Any one of the above descriptive characteristics or a combination of therein can be used. In cases where a large number of datasets are available to begin with, the datasets chosen to create the ontology need to come from diverse data sources and should encompass a wide array of possible column names in

order be effectively used or crosswalking new datasets. The standard schema can further be refined by standardizing column name formats, removing erroneous data and eliminating duplicate values.

Approach 1: Using Levenshtein Distance for Entity Resolution Model. The first step to be taken for this approach is the calculation of Levenshtein distance- a metric for measuring the distance between two metadata strings. The score ranges between 0 and 100, with 100 being an indication for exact match between the column strings. In addition to deciding on the method for derivation of the matching score, the minimum matching score that indicates a “qualified match” needs to be chosen (see Fig. 4). In other words, the threshold score, crossing which a pair of strings can be considered to be “closely matching” needs to be set. This naive approach is likely to work well for column metadata match but does not always capture context of inter-column relationships and ontology in complex cases.

The Levenshtein Distance between two strings is calculated as the number of changes needed to convert one string to another. The changes can be of three types: Insertion, Substitution or Deletion. The Levenshtein Distance between ELEPHANT and RELEVANT can be calculated as follows

Insertion (i)		E	L	E	P	H	A	N	T
Substitution (s)	d				s	i			
Deletion (d)	R	E	L	E	V		A	N	T

Since three changes have been made to ELEPHANT to convert it into to RELEVANT, the Levenshtein distance between the two strings is equal to 3

The matching score is calculated using the Levenshtein Distance using the following formula

$$\frac{\text{Length of Str1} + \text{Length of Str2} - \text{Levenshtein Distance}}{\text{Length of Str1} + \text{Length of Str2}} = \frac{8 + 8 - 3}{8 + 8} = 81.25$$

Fig. 4. Using Levenshtein distance

Approach 2: Using db2vec Method for Embeddings. The Db2Vec embedding from cognitive database systems, as researched by Boardwekar et al. [4] captures the inter column and intra column relationship between tables by treating them as an unstructured set. The first step in this process is the textification of the standard schema ontology. When the standard schema is provided as an input in the form of a data frame, the algorithm first textifies the data (using a preprocessing script to convert every row of the table into the required format sentences). The resulting textified standard schema is then used to train the (db2vec) model (see Fig. 5). The model takes the textified data as input and

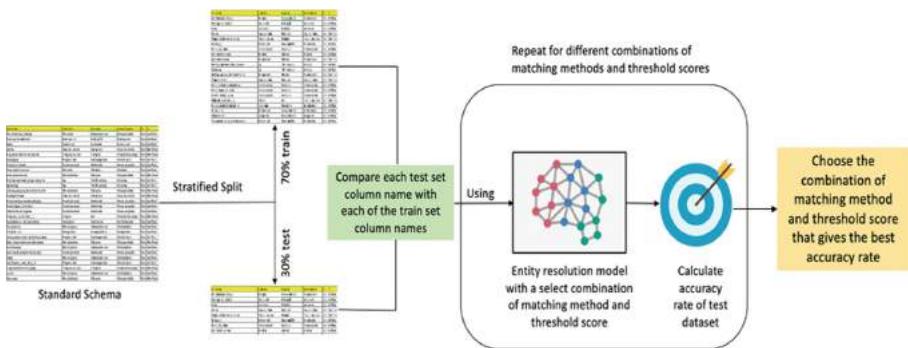


Fig. 5. Using db2vec method for embeddings

outputs a vector representation of the unique tokens in the textified data. The training involves the use of a highly specialized 3-layer Neural Network Model to generate vectors. Once trained, this model can take an unknown string as an input and provide a list of records from the standard schema that closely match the input string. In this use case, the input string is a column name from a newly ingested dataset. This model is adept at finding similar records even when there are errors or spelling mistakes in the input column. Moreover, the algorithm can be tuned to output a given number of similar records. When this number is set to 1, the model outputs a record that matches closest with the input column name.

The train set can now be replaced with the complete standard schema, and metadata of a new, unseen dataset can now take place of the test set. The model would compare each column metadata of the new dataset against the standard schema to find the closest match. The ontological structure assigned to the best matching column name would then be picked from the standard schema and assigned to the new column, thereby, rendering the model the capability to perform metadata crosswalking.

4 Experimentation

The aforesaid approaches have been experimented on the open data systems for Marine Litter managed by United Nations Environmental Program (UNEP) and Earth Challenge 2020. All these datasets [6–8] have different data collection methodology, different schemas and are not interoperable (see Fig. 6). A team of Data Stewards had spent several days of manual effort to harmonize metadata and created a interoperable dataset file with uniform metadata [9]. The methods outlined in this study have been tested on these source sets and the results were compared to the manually curated metadata (see Fig. 7).

Name	EC2020	T1	T2	T3	CSIRO	
Food_Wrappers_candy_chips_et	Wrapper or label	Plastic	Soft Plastic	Other soft plastic	Food wrapper/label	
Take_Out_Away_Containers_Plast	Food container	Plastic	Hard Plastic	Hard food container	Food container	
Take_Out_Away_Containers_foam	Food container	Plastic	Hard Plastic	Hard food container	Food container	
Bottle_Caps_Plastic_	Bottle cap or lid	Plastic	Hard Plastic	Bottle cap or lid	Bottle cap/lid	
Lids_Plastic_	Bottle cap or lid	Plastic	Hard Plastic	Bottle cap or lid	Bottle cap/lid	
Straws_Stirrers	Straw	Plastic	Soft Plastic	Straw	Straw	
Forks_Knives_Spoons	Plates, bowls, cups, or silver	Plastic	Hard Plastic	Hard utensil, plate, or bowl	Utensil/plate/bowl	
Beverage_Bottles_Plastic_	Beverage bottle	Plastic	Hard Plastic	Beverage bottle	Beverage bottle <1 L	
Grocery_Bags_Plastic_	Bag	Plastic	Soft Plastic	Grocery bag	Thin film carry bag	
Other_Plastic_Bags	Bag	Plastic	Soft Plastic	Grocery bag	Thin film carry bag	
Cups_Plates_Plastic_	Plates, bowls, cups, or silver	Plastic	Hard Plastic	Hard utensil, plate, or bowl	Utensil/plate/bowl	
Cups_Plates_Foam_	Plates, bowls, cups, or silver	Plastic	Hard Plastic	Hard utensil, plate, or bowl	Utensil/plate/bowl	
F6_Pack_Holders	String, ring, or ribbon	Plastic	Other Plastic	String, ring, or ribbon	String/rope/ribbon	
Other_Plastic_Foam_Packaging	Other plastic	Plastic	Hard Plastic	Other plastic debris	Unknown/other hard	
Other_Plastic_Bottles_oil_ble	Other bottle	Plastic	Hard Plastic	Other bottle	Other bottle	
Strapping_Bands	Packing strap	Plastic	Other Plastic	Straps, ties, or bands	Packing strap	

Fig. 6. Example of source schema sets

Example- For marine litter, the classification of plastic litter alone can be done in many ways by various schemas, requiring manual analysis to map them with each other. In the schema indicated in Fig. 6 T1, T2, T3 represent different ontological levels for same object (see Fig. 8).

```
document_vector_prediction(modified_text_data_split, "Used plates", column="NAME", method="idfbm25", total_num=5, \
type_col="TIER1")  
  
array(['TIER1!!METAL', 'TIER1!!OTHER', 'TIER1!!GLASS', 'TIER1!!PROCESSED',  
'TIER1!!LUMBER', 'TIER1!!TEXTILES', 'TIER1!!CLOTH',  
'TIER1!!PLASTIC', 'TIER1!!RUBBER'], dtype=object)
```

Fig. 7. Predicting column metadata

The test result indicated in Fig. 7 shows the output of the db2vec model [4]. For purpose of the experiment, only the column names and tier 1 (T1) has been provided as the input schema in the form of a dataframe. The algorithm first textified the input using a preprocessing script and every row of the dataframe is converted into the required format sentences. The resulting textified standard schema was then used to train the db2vec model. The model provided a vector representation of the unique tokens in the textified data using a robust 3-layer Neural Network Model. To test the model, list of strings (columns) from a test dataset have been provided as inputs. The model parameters have been set to output a list of 5 closely matching ‘Tier1’ values. An example has been presented in Fig. 7: when ‘Used Plates’ was provided as the test string, the Tier1 (T1) value was correctly classified as ‘Metal’. The model was also tested on completely new schemas. An accuracy of 82% was obtained a dataset consisting of few hundred columns in each of the child schemas.

Figure 8, shows how the current study can be used to derive an ontological structure for a new schemas. In this case, the complete schema structure indicated in Fig. 6 was provided as input, and the model predicted the ontological hierarchy for a previously unseen string ‘straws and bits’ correctly.

```

0 {
1   "predictions": {
2     {
3       "results": {
4         "EC2020": "Straw",
5         "Tier1": "Plastic",
6         "Tier2": "Soft Plastic",
7         "Tier3": "Soft Plastic",
8         "CSIRO": "Straw"
9       }
10    }
11  }
12 }

```

Fig. 8. Predicting ontology

5 Limitations

With the proposed approaches being ML-based in nature, the absence of proper training set can affect the accuracy of the crosswalking. Though these approaches lead to a fair degree of automation of the metadata harmonization and ontology derivation processes, they require some manual intervention, atleast at the beginning, in order to enable the models to work with better accuracy- for instance, the mappings automatically performed by the model need to be manually reviewed and corrected initially so that the models can learn from feedback from Data Steward and historical training data to become better. Complete lack of training data will make this approach harder to work.

6 Conclusion

This paper presents a novel approach to harmonize metadata automatically by creating a machine generated crosswalk. It further demonstrates that it can work in the absence any training data using contextual embeddings of entities, and also that the system can derive not just column names but also the ontological structure of schema automatically. The current research, thus, can aid in speeding up the pipeline, resulting in improved efficiency of the data curation process.

Statement on Impact

This research is primarily aimed at influencing the efficiency of ML-ops process in large enterprises. As popular wisdom goes; 80% of Machine Learning project effort is spent just on data engineering and within that, majority of effort is on cataloguing datasets for data curation, that is, for metadata harmonization across various schemas. Application of ML for improving ML lifecycle for the benefit of data stewards who are mainly responsible the manual work in data curation processes is the main intent of the authors. This will also benefit data engineers and data scientists by reducing the time need to gain access to the

right data. Authors currently do not foresee disadvantages of this method to any particular group nor see the implications of bias. Authors do understand that this research may change the nature of job for the community of Data Stewards but this should be viewed as advantageous to them by improving productivity of tasks instead of being a disadvantage or potential risk to job. Authors welcome any constructive feedback on the impact of this research which they may not have anticipated.

References

1. Wurzer, J.: Automated harmonization of data (2013)
2. Schon, A.: Matching metadata sources using rules for characterizing matches (2011)
3. Cope, B.: Method and apparatus for the creation, location and formatting of digital content (2003)
4. Bordawekar, R., Bandyopadhyay, B., Shmueli, O.: Cognitive database: a step towards endowing relational databases with artificial intelligence capabilities (2017)
5. Cobbett, M., Limburn, J., Oberhofer, M., Schumacher, S., Woolf, O.: Automatic Ontology generation (2020)
6. [Data] Trash Information and Data for Education and Solutions (TIDES): Plastic Pollution. <https://cscloud-ec2020.opendata.arcgis.com/datasets/data-marine-litter-watch-mlw-plastic-pollution-/explore>
7. [Data] Marine Litter Watch (MLW): Plastic Pollution. <https://cscloud-ec2020.opendata.arcgis.com/datasets/data-marine-litter-watch-mlw-plastic-pollution-/explore>
8. [Data] Marine Debris Monitoring and Assessment Project (MDMAP) Accumulation Report: Plastic Pollution. <https://cscloud-ec2020.opendata.arcgis.com/datasets/data-marine-debris-monitoring-and-assessment-project-mdmap-accumulation-report-plastic-pollution>
9. [Data] Earth Challenge Integrated Data: Plastic Pollution (MLW, MDMAP, TIDES). <https://cscloud-ec2020.opendata.arcgis.com/datasets/data-earth-challenge-integrated-data-plastic-pollution-mlw-mdmap-tides>



Data Segmentation via t-SNE, DBSCAN, and Random Forest

Timothy DeLise^(✉)

Université de Montréal, Montreal, Canada
timothy.delise@umontreal.ca

Abstract. This research proposes a data segmentation algorithm which combines t-SNE, DBSCAN, and Random Forest classifier to form an end-to-end pipeline that separates data into natural clusters and produces a characteristic profile of each cluster based on the most important features. Out-of-sample cluster labels can be inferred, and the technique generalizes well on real data sets. We describe the algorithm and provide case studies using the Iris and MNIST data sets, as well as real social media site data from Instagram. This is a proof of concept and sets the stage for further in-depth theoretical analysis.

Keywords: Clustering · t-SNE · Random forest · DBSCAN · Segmentation · Data visualization · Unsupervised learning

1 Introduction

Data segmentation refers to the process of dividing data into clusters and interpreting the characteristics of these clusters, which information can be used for decision making purposes. It is clustering but with an additional requirement to understand the reason behind the clustering and stratification of the data. Data segmentation is widely used in a broad range of fields from social media site marketing [16] to the analysis of single-cell RNA sequencing [7,9]. There are many possible choices of clustering technique as well as possible methods of interpreting the characteristics of each cluster. This research proposes an intuitive, general purpose data segmentation technique which delivers interpretable clusters and tends to generalize well. The algorithm, pictured in Fig. 1, is comprised of three main steps: t-SNE, DBSCAN, and Random Forest classifier. In the text, this process is referred to simply as *the algorithm*.

T-distributed Stochastic Neighbor Embedding (t-SNE) is the basis of the clustering method. It has been chosen because of its vast popularity in the natural sciences [7,8,14], and it is widely regarded as the state of the art for dimension reduction for visualization [12]. T-SNE creates a low-dimensional embedding of high-dimensional data with the ability to retain both local and global structure in a single map. It has proven successful for visualizing high-dimensional data [17]. There is strong evidence to support that t-SNE embeddings recover well-separated clusters from the input data [11]. In practice, high-dimensional data

tends to produce distinctly isolated clusters by visual inspection of the low-dimensional output embedding [7]. The motivation is to harness the intuitive appeal of the t-SNE embedding. However, t-SNE by itself does not label clusters nor provide information about how and why the clusters appear. Moreover, an aspect of t-SNE that detracts from its ability for inference is that there is no direct map from the input space to the output embedding.

The algorithm harnesses t-SNE as an intuitive first step to simply visualize the data. Its great appeal is that we can visually inspect a low-dimensional embedding (an image, for example) and manually pick out clusters. In order to automate this process, we use DBSCAN [4] to extract clusters directly from this low-dimensional embedding. The reason for choosing DBSCAN, as opposed to other density-based clustering algorithms, is that it has a small number of important parameters and is foundational in the field of density-based clustering. The task of extracting visually-identifiable clusters from data in the plane is something that the DBSCAN algorithm can confidently accomplish.

There is one important parameter of the DBSCAN algorithm, defined in the reference as the *Eps-neighborhood* of a point p : $N_{eps}(p)$, that we will simply call ϵ . For specific data, it is possible to select a value for ϵ that separates dense regions into clusters. Through cross validation, optimal ϵ values can be discovered. In fact, tuning ϵ can help recover clusters at different levels of resolution.

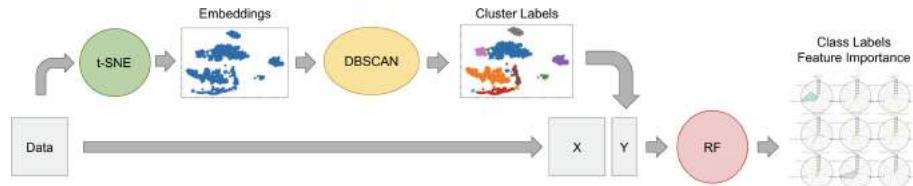


Fig. 1. This flowchart shows the design of the segmentation analysis algorithm. The data we assume is organized in the usual manner with rows representing individual data points and the columns representing the features of the data. The t-SNE algorithm is applied to the data resulting in a 2-dimensional embedding. Then the DBSCAN algorithm is applied to this embedding, resulting in labeled clusters of the data. Finally these labeled clusters are used as target labels for the random forest classifier, using the original (high dimensional) data as input. The random forest then has the ability to map data points to cluster labels, and also gives access to feature importance scores.

The third and final step of the algorithm uses the cluster labels obtained from the DBSCAN algorithm to train a Random Forest Classifier [2]. The utility of the Random Forest is two-fold: to infer cluster labels directly from the input data, and to gain access to feature importance scores. Random Forest was chosen as well because of its strong ability to classify data, especially if we have reason to believe the target classes belong to well-separated data points. Moreover, it gives transparency to the question of how the data is separated in the input space via

its feature importance scores. This fulfills interpretability requirement of data segmentation.

An important question to investigate is *how well does the algorithm generalize?* While data segmentation is inherently an unsupervised learning technique, not concerned with ground-truth data labels, there is a way to understand its ability to generalize. Simply put, if we separate our data set into training and test sets, then the algorithm applied to the training set should create the same clusters as the model applied to the entire set. We then compare the cluster labels given to the test set, using the Random Forest from the algorithm trained on the training data, to the cluster labels of the test set given by the algorithm trained on the entire data set. If the algorithm tends to generalize well on experimental data sets. This means that cluster labels of out-of-sample data points can be reliably inferred without retraining the model. It lends evidence that we can trust the feature importance scores of the Random Forest, which describe how and why the clusters are formed.

The structure of this paper is outlined as follows. Section 2 provides the details of the algorithm as well as the technique to assess its generalizability. Section 3 describes empirical examples of the algorithm applied to three experimental data sets: the Iris data set, a data set of anonymized Instagram data obtained from previous research [6], and the MNIST data set of 70,000 handwritten digits. Section 4 offers interpretation of the results and motivation for future research.

2 Methods

2.1 The Algorithm

The algorithm is composed of three main sub-algorithms: t-SNE, DBSCAN, and Random Forest classifier. Figure 1 gives an overview. We assume that the data is in the usual format, with rows representing individual data points and columns representing the features. T-SNE creates a 2-dimensional embedding of the data. For the next step, the DBSCAN algorithm is applied to the low-dimensional embedding to produce cluster labels for each data point. Finally these cluster labels are used to train a Random Forest classifier via supervised learning. The Random Forest model can thus infer cluster labels directly from the raw input data.

Certain values for ϵ reveal the clusters which are visually apparent in the t-SNE embedding. Most values of ϵ generalize well, although values for ϵ can be found that generalize extremely well, almost perfectly. In practice we optimize a constant, which is then multiplied by the mean pairwise distance of the t-SNE embedded data points. For more information about ϵ tuning via cross validation, please refer to Sect. 2.3.

The Random Forest admits feature importance scores. These scores allow us to understand which features are most influential in separating the data into clusters. Combining these scores with *cluster profiles* completes the process of segmenting the data, and hence the algorithm.

2.2 Cluster Profiles

We define the *cluster profile* to be the distribution of the data points of each feature over that cluster, as in Fig. 3. A simple statistic is the mean value. If our input data has n features, then the cluster profile can be represented is an n dimensional vector of the mean values of each cluster, as shown in Table 2 and Fig. 7. The cluster profile is thus used to characterize the cluster. The feature importance scores of the Random Forest algorithm allow us to focus on the features which matter the most. For example, we quickly understand that petal length is much more important than sepal width, for the purposes of dividing the iris data into clusters (Table 1).



Fig. 2. The plot on the left is the 2-dimensional embedding that resulted from the t-SNE part of the algorithm. The plot on the right shows the same data points labeled by the cluster labels that were learned using the DBSCAN algorithm applied to the Embedding.

Table 1. Iris data set feature importance scores calculated by the random forest classifier.

Score	Feature name
0.555780	Petal length (cm)
0.314322	Petal width (cm)
0.122197	Sepal length (cm)
0.007701	Sepal width (cm)

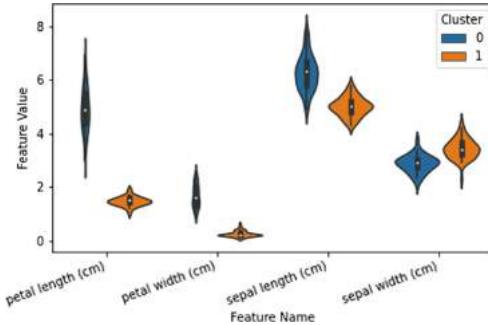


Fig. 3. The Iris cluster profiles are shown in a *Violin Plot*, which displays the empirical distribution of the data over each feature, separated by cluster.

2.3 Generalizing Segments

Cluster profiles are developed and we wish for these clusters and their characteristics to generalize to out-of-sample data points. The algorithm gives a way to infer cluster labels of out-of-sample data points using the Random Forest classifier. Here, we describe a technique for assessing the generalizability of the algorithm.

In unsupervised learning scenarios, the data does not contain ground-truth labels, so we take the ground-truth of some particular data point to be the cluster label that is assigned to that data point when the entire data set is run through the algorithm. We then randomly split the whole data set into training (in-sample) and test (out-of-sample) sets in the usual way. In our case we use the 5-fold cross validation technique described in [3]. For each fold of data, the algorithm is run on the training set, which returns the cluster labels of the training data and a Random Forest classifier that will map input data to cluster labels. Finally we infer the cluster labels from the test set by applying the Random Forest classifier. Classification metrics are computed using the labels obtained from the test set compared to the *ground-truth* labels that were computed from the entire data set. In Table 4 the weighted averages of the classification metrics over all 5 folds of the data are displayed.

Cross validation is used, as well, on each training set to choose the optimal value of the DBSCAN parameter ϵ . This makes the generalization procedure quite computation-intensive since the training set for each fold of the 5-fold cross validation is used to optimize ϵ by way of 5-fold cross validation. This additional computation time is merited for the purpose of a thorough analysis of the algorithm. In practice the ϵ parameter can be chosen using less expensive means and validated using cross validation before applying it to the entire data set. We forego the cross validation of the ϵ parameter in the MNIST experiment for the sake of time savings and additional *resolution* of the clusters.

The idea of cluster resolution can be illustrated by considering the following thought experiment. A very large value for ϵ will always produce only one cluster,

and this technique will obviously always generalize perfectly. Depending on the data, we may wish to set an lower limit to the number of clusters obtained, thus sacrificing performance for the sake of segmenting the data into more, smaller clusters. This purpose is inherently attained by selecting smaller values for ϵ . By lowering ϵ we derive more clusters from the data, but this also creates more singleton (and extremely small) clusters which detract from the generalizing performance. For the Iris data set we use a lower limit of clusters we require to 2, and for the Instagram data we set the limit to 5. In the MNIST data experiment we intentionally set ϵ small enough to reveal the 10 main clusters of the data, therein creating many small and singleton clusters.

The cluster labels obtained from the analysis of the training data set need not match the labels obtained from the whole data set. The reason is that the cluster label name is chosen somewhat arbitrarily in that we always label the largest cluster as cluster 0, the next largest cluster as cluster 1 and so on. In fact, it is common for the training set to produce a different number of clusters than the whole data set altogether. We have developed a technique to address this by matching the clusters obtained from the training set with those from the entire data set. It is an iterative procedure that matches clusters which have the largest intersection first. Details about this procedure are supplied in Appendix A.

An alternative technique to compute the out-of-sample classification metrics is to map out-of-sample data points to their embedded location, something that is not possible in the original t-SNE algorithm, however has been implemented in the openTSNE software package [15]. We chose not to use this technique in order to focus on the utility of the Random Forest step of the algorithm. However, one should be able to show similar results using the inference mapping of openTSNE.

2.4 Software

The software used for the experiments will be made freely available on GitHub. It is a conglomerate of customized code and algorithms with existing software packages. Scikit-learn [13] was used for the Random Forest and DBSCAN implementations as well as data scaling and classification metrics. FIt-SNE [10] was used for the t-SNE computations as it is fast and has shown success visualizing the MNIST data set.

Table 2. Instagram data set cluster profiles. For each of the five clusters we display the mean values of the top five important features over each cluster.

Cluster	Follows	Average shortest path	Diameter	Clique count	Node count
0	38.07	186.98	126.41	1.86	0.63
1	345.61	585.47	587.62	1.90	0.68
2	0.13	0.00	0.00	0.00	0.00
3	113.09	264.17	180.31	0.98	0.61
4	122.24	328.97	260.71	1.30	0.63

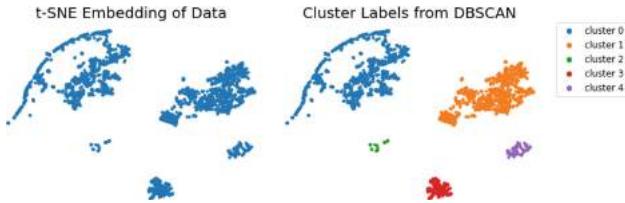


Fig. 4. The plot on the left shows the 2-dimensional embedding of the Instagram data set that resulted from t-SNE. On the right side is the same embedding with the cluster labels given by the DBSCAN step.

Table 3. Instagram data set feature importance table, showing the top ten most important features, ordered by score.

Score	Feature name
0.110567	Follows
0.100257	Average shortest path
0.091884	Diameter
0.080912	Clique count
0.057833	Node count
0.054657	Followed by
0.049908	Follow ratio
0.046561	Edge connectivity
0.045851	Edge count
0.044900	Node connectivity
0.044383	Average connectivity

3 Empirical Results

3.1 Iris Data Set

The Iris flower data set [1,5] is a famous, elegant and freely available data set that displays intrinsic clusters. Figure 2, 3 and Table 1 display the output of the algorithm applied to the Iris flower data set. There are clearly 2 clusters in the data. Table 4 shows the classification metrics for each generalization experiment.

Referring to Fig. 3, the goal is to understand why the constituents of each cluster have been grouped together. The clusters label is assigned by the number of data points in each cluster. We see that cluster 0 is characterized by longer petal length, petal width, and sepal length than cluster 1 while having

shorter sepal width. This simple visualization tool, while by no means exhaustive, already offers substantial insight into the descriptive attributes of each cluster. There is very little overlap between the clusters in the distributions of petal length and width. We understand that petal length and width are more important for inferring these clusters than sepal length and width. This idea matches precisely with the feature importance scores of Table 1. Those familiar with the data set will know that these are measurements from three types of flowers: Iris Setosa, Iris Versicolor, and Iris Virginica. The measurements from Versicolor and Virginica tend to mix while the Setosa is quite separate. It corresponds that the segmentation analysis was able to identify two clusters and not three.

3.2 Instagram Data

The analysis of this section follows the same steps as the previous section, the only difference being that we substitute the input data. The data was obtained from a previous study [6] and is completely anonymized. The features contain simple metrics about Instagram users, such as the number of followers, likes, tags, etc. We also calculated several social network attributes based on the raw data. This data set contains 3,229 data points and 27 features. For more information about the data, please refer to [6].

Figure 4 shows the clustering results from the segmentation analysis. The feature importance scores of Table 3 combined with the cluster profiles of Table 2 give us the defining characteristics of the clusters.

Cluster 0 is the largest cluster and cluster 1 is the next largest, corresponding to the blue and orange clusters of Fig. 4, respectively. Cluster 0 is described by data points with less follows, average shortest path, and diameter, and cluster 1 has higher values for these important features. We see cluster 2, which is the third largest cluster, has a mean that is zero or almost zero across the important features. These are seemingly empty accounts. The segmentation analysis makes it easy to understand how the data is stratified.

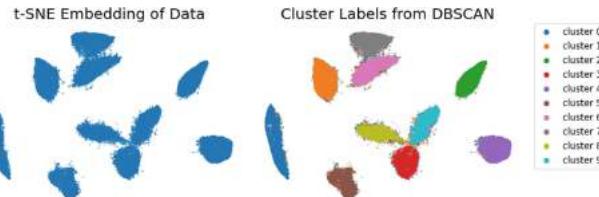


Fig. 5. MNIST database t-SNE embedding and top ten derived clusters. Notice the many small sporadic clusters that are produced around the edges of the main clusters.

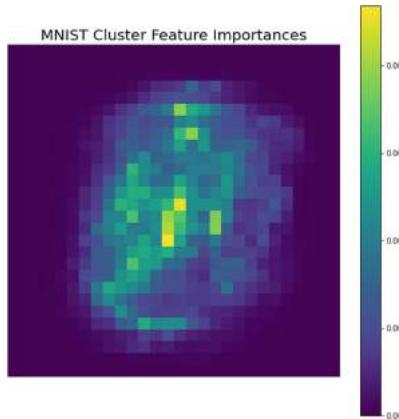


Fig. 6. A heat map of the feature importance scores learned by the random forest step of the algorithm applied to the MNIST data set. In this experiment, features correspond to pixels, so we display the feature scores that correspond to each pixel. We notice that the important features are located toward the center of the image, which is the area of the image where the digits appear.

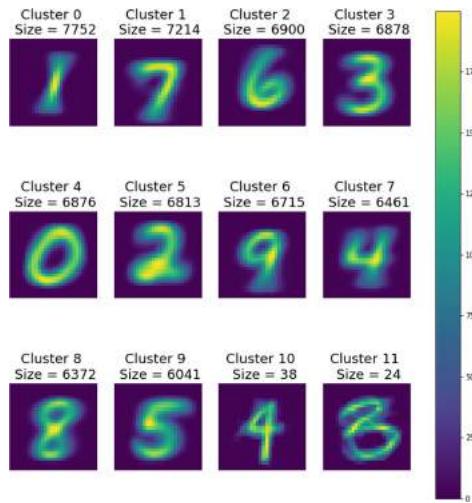


Fig. 7. The cluster profiles for the top 12 clusters (by Size) derived from the MNIST data set. The cluster profile is the mean of each feature over the cluster. Just like Fig. 6, the features correspond to pixels, so the cluster profiles are displayed as images, where each pixel is the mean of all the respective pixels from each cluster. We notice that the first 10 clusters are all substantially larger than the remaining clusters. This is also apparent from the embedding image of Fig. 5. Each of the largest ten clusters are representative of each of the ten digits.

3.3 MNIST Case Study

There are, inevitably, settings for which the default parameters for t-SNE don't quite get us the best embedding. Tuning t-SNE can sometimes produce better visual clusters. The purpose of this section is to illustrate that the algorithm is robust in regards to parameter tuning. Here we address the MNIST data set of 70,000 hand-written images. The embedding produced using the default parameters for t-SNE does not clearly separate ten clusters. However, by using late exaggeration [9], the authors of [10] show that clusters clearly appear in the produced embedding. Although this data set contains 10 distinct data labels corresponding to each of the first ten digits, it has traditionally been difficult for clustering algorithms to clearly identify clusters corresponding to these ten digits.

Even though the embedding on the left side of Fig. 5 seems to show ten distinct clusters, a few of the clusters are slightly touching in certain regions. ϵ has been adjusted in order to capture the ten main segments of the data. In doing so a bit of performance was sacrificed in that many very small clusters, often singleton clusters, were identified, which are very difficult to generalize. Nevertheless, we found that the clusters identified in this way still generalize well by weighted average measure.

The feature importance scores highlight the important features that contribute to the separation of the clusters. Since each feature corresponds to a pixel, this conveniently gives an intuitive interpretation where we can visualize the important pixels spatially on a two dimensional image in Fig. 6. The result agrees with our intuition that the middle section of the image should be most important for separating the data into clusters.

Finally, we visualize the cluster profiles in Fig. 7 in image form as well. The ten largest clusters, in fact, correspond to representations of the ten digits. By focusing on the largest clusters and the most important features, we can understand a vast majority of the data.

3.4 Generalization Performance

Table 4. The classification metrics for each of the experiments use the weighted average method for calculations. These numbers represent the mean of each weighted score across all five folds of the data.

Data set	Accuracy	Precision	Recall	F1-score
Iris	1	1	1	1
Instagram	0.943	0.996	0.943	0.967
MNIST	0.916	0.918	0.916	0.911

Table 4 displays the accuracy, precision, recall and f1-score as a weighted average over all five folds of the data. We find for these data sets, the algorithm generalizes well in the sense that out-of-sample data points are most likely going to be

classified into the correct cluster by the Random Forest classifier. The good performance emphasizes our belief that the feature importance scores produced by the Random Forest are useful. Moreover, we are confident that the information gained from the algorithm extends to a broader population.

4 Conclusion and Future Work

It deserves to be written that this research is superficial in nature and relies on statistical evidence as a proof-of-concept. The author intends to further develop a theoretical understanding. The value of this paper is for the engineer or data science practitioner who needs to get answers from data for which there is little understanding. It relies on the success of t-SNE for visualizing data and adds a layer of interpretability in a practical sense. The additional step is subtle but important for mission-critical applications.

There are some theoretical connections to be made between the t-SNE and DBSCAN step. One of the main results of [11] is a theoretical guarantee that all clusters of the input data will be mapped to *balls* in the embedding which can be made arbitrarily small. It is plausible that DBSCAN can successfully identify such *balls* in a low-dimensional space based on the ideas of connectivity and reachability in density-based clustering. The remaining piece is to create a formal argument that if the DBSCAN algorithm has identified a cluster in the embedding space, then this must correspond to a cluster in the input space. This will be a topic of future research.

Random Forest has been a robust supervised learning tool for a long time. If we guarantee that we have labeled actual clusters in the input data, which should follow from the previous paragraph, then we should expect the Random Forest classifier to be able to successfully classify these data points. There should be a way to statistically guarantee that Random Forests can classify disjoint clusters of data. This is another direction of future research. The outline given in these two paragraphs should deliver a more substantial theoretic argument for why this algorithm can dependability be used for data segmentation.

Appendix

A Matching Clusters for Generalization Analysis

This section describes the algorithm used to match clusters between the entire data set and the training data sets that are split during each fold of the generalization analysis. As mentioned in the text, we perform the equivalent of 5-fold cross validation to calculate the average weighted f1-score across all 5-folds of the data. During each fold, we needed a technique to pair the clusters derived from training data with the clusters from the entire data set. This comes down to matching cluster labels, since the labels assigned to each cluster do not necessarily match between runs of the algorithm.

The effect of this matching is really very subtle. Let us do a simple thought experiment by considering the Iris data set, where we saw two main clusters. Something that could happen is that the clusters derived from the entire data set are labeled cluster 0 and cluster 1. The clustering results from the training data during one of the folds of could have derived two main clusters as well, however the algorithm could have labeled cluster 0 as cluster 1, and cluster 1 as cluster 0. The matching outlined in this section simply gives us a quick technique to match those labels.

The technique here is also robust to the situation where the training data derives a different number of clusters than the entire data set. Algorithm 1 will match as many clusters as it can, in a largest-first fashion. The optimal algorithm would consider all the permutations of the clusters of the training data compared to the entire data set, aiming to maximize the intersection of all the clusters, however this can require too many computations on large data sets. We sacrifice a bit of performance in terms of f1-score in lieu of considerable time benefits.

Algorithm 1: Algorithm for Matching Cluster Labels Between Entire Data Set and Training Data Set

Result: BestPerm is list that maps the cluster labels from the AllClusters to TrainClusters. The index position of BestPerm corresponds to the cluster label number of TrainClusters, and the value in that position corresponds to the cluster label of AllClusters.

TrainClusters is a list of clusters from training data;

AllClusters is a list of clusters from the entire data set;

for *Cluster1* in *TrainClusters* **do**

 BestSum = 0;

 BestCluster = None;

for *Idx*, *Cluster0* in *AllClusters* **do**

 ThisSum = Size of Intersection of *Cluster1* and *Cluster0*;

if *ThisSum* is greater than *BestSum* and *Idx* not in *BestPerm* **then**

 | BestSum = *ThisSum*;

 | BestCluster = *Idx*;

 | **end**

 | **end**

 | BestPerm.append(BestCluster);

end

References

1. Anderson, E.: The species problem in Iris. Ann. Mo. Bot. Gard. **23**(3), 457–509 (1936)
2. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>

3. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees, 1st edn. Wadsworth Inc., Belmont (1984)
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, pp. 226–231. AAAI Press (1996)
5. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **7**(7), 179–188 (1936)
6. Hassanpour, S., Tomita, N., DeLise, T., Crosier, B., Marsch, L.A.: Identifying substance use risk based on deep neural networks and Instagram social media data. *Neuropsychopharmacology* **44**(3), 487–494 (2019)
7. Kobak, D., Berens, P.: The art of using t-SNE for single-cell transcriptomics. *Nat. Commun.* **10**(1), 5416 (2019)
8. Li, W., Cerise, J.E., Yang, Y., Han, H.: Application of t-SNE to human genetic data. *J. Bioinform. Comput. Biol.* **15**, 06 (2017)
9. Linderman, G.C., Rachh, M., Hoskins, J.G., Steinerberger, S., Kluger, Y.: Efficient algorithms for t-distributed stochastic neighborhood embedding. CoRR, abs/1712.09005 (2017)
10. Linderman, G.C., Rachh, M., Hoskins, J.G., Steinerberger, S., Kluger, Y.: Fast interpolation-based t-SNE for improved visualization of single-cell RNA-SEQ data. *Nat. Methods* **16**(3), 243–245 (2019)
11. Linderman, G.C., Steinerberger, S.: Clustering with t-SNE, provably. CoRR, abs/1706.02582 (2017)
12. McInnes, L., Healy, J., Melville, J.: Umap: uniform manifold approximation and projection for dimension reduction (2020)
13. Pedregosa, F.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
14. Platzer, A.: Visualization of SNPs with t-SNE. *PloS ONE* **8**(2), e56883 (2013)
15. Poličar, P.G., Stražar, M., Zupan, B.: openTSNE: a modular python library for t-SNE dimensionality reduction and embedding. bioRxiv (2019)
16. Rajagopal, S.: Customer data clustering using data mining technique. *Int. J. Database Manag. Syst.* **3**, 12 (2011)
17. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008)



GeoTree: A Data Structure for Constant Time Geospatial Search Enabling a Real-Time Property Index

Robert Miller^(✉) and Phil Maguire

National University of Ireland, Maynooth, Kildare, Ireland
`{robert.miller,phil.maguire}@mu.ie`

Abstract. A common problem appearing across the field of data science is k -NN (k -nearest neighbours), particularly within the context of Geographic Information Systems. In this article, we present a novel data structure, the GeoTree, which holds a collection of geohashes (string encodings of GPS co-ordinates). This enables a constant $O(1)$ time search algorithm that returns a set of geohashes surrounding a given geohash in the GeoTree, representing the approximate k -nearest neighbours of that geohash. Furthermore, the GeoTree data structure retains an $O(\cdot)$ memory requirement. We apply the data structure to a property price index algorithm focused on price comparison with historical neighbouring sales, demonstrating an enhanced performance. The results show that this data structure allows for the development of a real-time property price index, and can be scaled to larger datasets with ease.

Keywords: GeoTree · Geospatial · k -NN · Data structure · Price index

1 Introduction

Large scale datasets are a hot topic in computer science. Each one tends to present its own problems and intricacies [9]. The Nearest Neighbour (NN) problem is a well known and vital facet of many data mining research topics. This involves finding the nearest data point to a given point under some metric which measures the *distance* between data points. In the context of geospatial data, the NN problem often emerges in the form of geographical proximity search [24].

Real world geographic data is usually represented by a pair of GPS co-ordinates, which pinpoint any location on Earth with unlimited precision. As a result of their structure, computing the distance between pairs of points in order to find the *nearest neighbour* can be extremely slow on large datasets.

The problem often requires expansion to finding the k nearest neighbours (k -NN), which further increases the complexity by requiring a sorting of the distance matrix in order to extract a ranking of points by proximity. It is extremely computationally expensive to compute and rank these distances on large datasets

[25]. A computationally cheap method of solving this problem would vastly improve the scalability of proximity based algorithms [24]. We propose a data structure which enables such cheap computation, the GeoTree, and explore its potential when applied to a real-world geospatial task.

2 Background

2.1 Naive Geospatial Search

The distance between two pieces of geospatial data defined using the GPS coordinate system is computed using the *haversine* formula [23]. If we wish to find the closest point in a dataset to any given point in a naive fashion, we must loop over the dataset and compute the haversine distance between each point and the given, fixed point. This is an $O(n)$ computation. If the distances are to be stored for later use, this also requires $O(n)$ memory consumption. Thus, if the closest point to every point in the dataset must be found, this requires an additional nested loop over the dataset, resulting in $O(n^2)$ memory and time complexity overall (assuming the distance matrix is stored). If such a computation is applied to a large dataset, such as the 147,635 property transactions used in the house price index developed by [14], an $O(n^2)$ algorithm can run extremely slowly even on powerful modern machines.

As GPS co-ordinates are multi-dimensional objects, it is difficult to prune and cut data from the search space without performing the haversine computation. With a considerable portion of big data being geospatial in nature, geospatial algorithms and data structures are coming under increased research attention, with the amount of personal location data available growing by approximately 20% year-on-year according to the *McKinsey Global Institute* [13]. As such, exploring alternative methods of representing GPS co-ordinates is necessary to make algorithmic improvements.

2.2 GeoHash

A geohash is a string encoding for GPS co-ordinates, allowing co-ordinate pairs to be represented by a single string of characters. The publicly-released encoding method was invented by Niemeyer in 2008 [20]. The algorithm works by assigning a geohash string to a square area on the earth, usually referred to as a *bucket*. Every GPS co-ordinate which falls inside that bucket will be assigned that geohash. The number of characters in a geohash is user-specified and determines the size of the bucket. The more characters in the geohash, the smaller the bucket becomes, and the greater precision the geohash can resolve to. While geohashes thus do not represent points on the globe, as there is no limit to the number of characters in a geohash, they can represent an arbitrarily small square on the globe and thus can be reduced to an exact point for practical purposes. Figure 1 demonstrates parts of the geohash grid on a section of map.

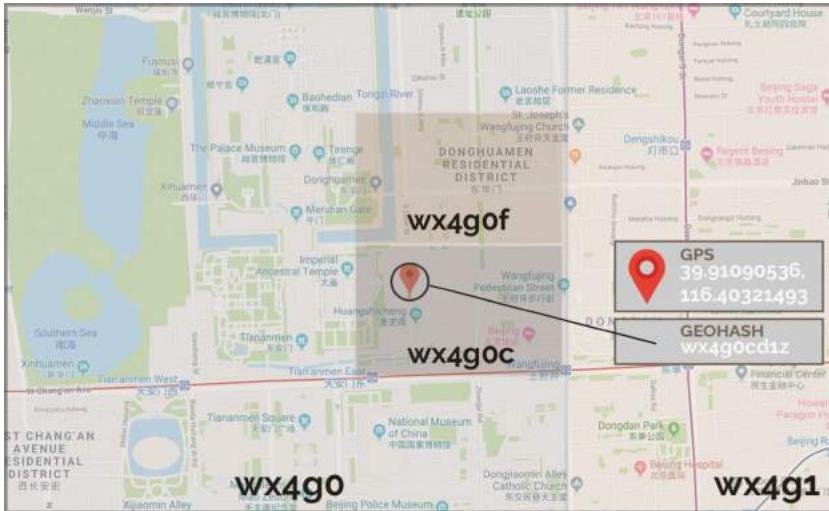


Fig. 1. GeoHash algorithm applied to a map

Geohashes are constructed in such a way that their string similarity signifies something about their proximity on the globe. Take the longest sequential substring of identical characters possible from two geohashes (starting at the first character of each geohash) and call this string x . Then x itself is a geohash (i.e. a bucket) with a certain area. The longer the length of x , the smaller the area of this bucket. Thus x gives an upper bound on the distance between the points. We will refer to this substring as the *smallest common bucket* (SCB) of a pair of geohashes. We define the length of the SCB as the length of the substring defining it. This definition can additionally be generalised to a set of geohashes of any size. Furthermore, we define the SCB of a single geohash g to be the set of all geohashes in the dataset which have g as a prefix. We can immediately assert an upper bound of 123,264 m for the distance between the geohashes in Fig. 2, as per the table of upper bounds in the *pygeohash* package [15].

$$\begin{aligned} \text{geohash 1: } & \underbrace{c_1 c_2 c_3}_{\text{SCB}} x_4 \dots x_n \\ \text{geohash 2: } & \underbrace{c_1 c_2 c_3}_{\text{SCB}} y_4 \dots y_n \\ \text{where: } & x_i \neq y_i \forall i \in \{4 \dots n\} \end{aligned}$$

Fig. 2. Geohash precision example

2.3 Efficiency Improvement Attempts

Geohashing algorithms have, over time, improved in efficiency and have been put to use in a wide variety of applications and research contexts [17, 18]. As stated by [24], the efficient execution of nearest neighbour computations requires the use of niche spatial data structures which are constructed with the proximity of the data points being a key consideration.

The method proposed by Roussopoulos et al. [24] makes use of *R-trees*, a data structure very similar in nature to the geohash [8]. They propose an efficient algorithm for the precise *NN* computation of a spatial point, and extend this to identify the exact k -nearest neighbours using a subtree traversal algorithm which demonstrates improved efficiency over the naive search algorithm. Arya et al. [2] further this research by introducing an approximate k -NN algorithm with time complexity of $O(kd \log n)$ for any given value of k .

A comparison of some data structures for spatial searching and indexing was carried out by [12], with a specific focus on comparison between the aforementioned *R-trees* and *Quadtrees*, including application to large real-world GIS datasets. The results indicate that the Quadtree is superior to the R-tree in terms of build time due to expensive R-tree clustering. As a trade-off, the R-tree has faster query time. Both of these trees are designed to query for a very precise, user-defined area of geospatial data. As a result they are still quite slow when making a very large number of queries to the tree.

Beygelzimer et al. [4] introduce another new data structure, the cover tree. Here, each level of the tree acts as a “cover” for the level directly beneath it, which allows narrowing of the nearest neighbour search space to logarithmic time in n .

Research has also been carried out in reducing the searching overhead when the exact k -NN results are not required, and only a spatial region around each of the nearest neighbours is desired. It is often the case that ranged neighbour queries are performed as traditional k -NN queries repeated multiple times, which results in a large execution time overhead [3]. This is an inefficient method, as the lack of precision required in a ranged query can be exploited in order to optimise the search process and increase performance and efficiency, a key feature of the GeoTree.

Muja et al. provide a detailed overview of more recently proposed data structures such as partitioning trees, hashing based *NN* structures and graph based *NN* structures designed to enable efficient k -NN search algorithms [19]. The *suffix-tree*, a data structure which is designed to rapidly identify substrings in a string, has also had many incarnations and variations in the literature [1]. The GeoTree follows a somewhat similar conceptual idea and applies it to geohashes, allowing very rapid identification of groups of geohashes with shared prefixes.

The common theme within this existing body of work is the sentiment that methods of speeding up k -NN search, particularly upon data of a geospatial nature, require specialised data structures designed specifically for the purpose of proximity searching [24].

3 GeoTree

The goal of our data structure is to allow efficient approximate ranged proximity search over a set of geohashes. For example, given a database of house data, we wish to retrieve a collection of houses in a small radius around each house without having to iterate over the entire database. In more general terms, we wish to pool all other strings in a dataset which have a maximal length SCB with respect to any given string.

3.1 High-Level Description

A GeoTree is a general tree (a tree which has an arbitrary number of children at each node) with an immutable fixed height h set by the user upon creation. Each level of the tree represents a character in the geohash, with the exception of level zero - the root node. For example, at level one, the tree contains a node for every character that occurs among the first characters of each geohash in the database. For each node in the first level, that node will contain children corresponding to each possible character present in the second position of every geohash string in the dataset sharing the same first character as represented by the parent node. The same principle applies from level three to level h of the GeoTree, using the third to h^{th} characters of the geohash, respectively.

At any node, we refer to the path to that node in the tree as the *substring* of that node, and represent it by the string where the i^{th} character corresponds to the letter associated with the node in the path at depth i .

The general structure of a GeoTree is demonstrated in Fig. 3. As can be seen, the first level of the tree has a node for each possible letter in the alphabet. Only characters which are actually present in the first letters of the geohashes in our dataset will receive nodes in the constructed tree. We, however, include all characters in this diagram for clarity. In the second level, the *a* node also has a child for each possible letter. This same principle applies to the other nodes in the tree. Formally, at the i^{th} level, each node has a child for each of the characters present among the $(i+1)^{th}$ position of the geohash strings which are in the SCB of the current substring of that node. A worked example of a constructed GeoTree follows in Fig. 4.

Consider the following set of geohashes which has been created for the purpose of demonstration: $\{gc7j98, gc7j98, gd7j98, ac7j98, gc9aaaj, gc7j9d, ac7j98, gd7jya, gc9aaaj\}$. The GeoTree generated by the insertion of the geohashes above with a fixed height of six would appear as seen in Fig. 4.

3.2 GeoTree Data Nodes

The data attributes associated with a particular geohash are added as a child of the leaf node of the substring corresponding to that geohash in the tree, as shown in Fig. 5. In the case where one geohash is associated with multiple data entries, each data entry will have its own node as a child of the geohash substring, as demonstrated in the diagram.

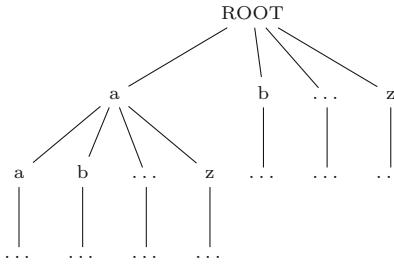


Fig. 3. GeoTree general structure

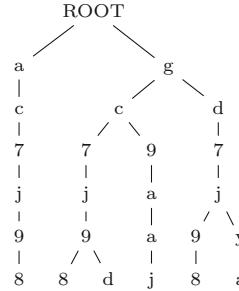


Fig. 4. Sample GeoTree structure

It is now possible to collect all data entries in the SCB of a particular geohash substring without iterating over the entire dataset. Given a particular geohash in the tree, we can move any number of levels up the tree from that geohash's leaf nodes and explore all nearby data entries by traversing the subtree given by taking that node as the root. Thus, to compute the set of geohashes with an SCB of length m or greater with respect to the particular geohash in question, we need only explore the subtree at level m along the path corresponding to that particular geohash. Despite this improvement, we wish to remove the process of traversing the subtree altogether.

3.3 Subtree Data Caching

In order to eliminate traversal of the subtree we must cache all data entries in the subtree at each level. To cache the subtree traversal, each non-leaf node receives an additional child node which we will refer to as the *list* (*ls*) node. The list node holds references to every data entry that has a leaf node within the same subtree as the list node itself. As a result, the list node offers an instant enumeration of every leaf node within the subtree structure in which it sits, removing the need to traverse the subtree and collect the data at the leaf nodes. The structure of the tree with list nodes added is demonstrated in Fig. 6 (some nodes and list nodes are omitted for the sake of brevity and clarity).

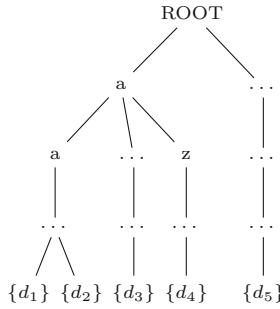


Fig. 5. GeoTree structure with data nodes

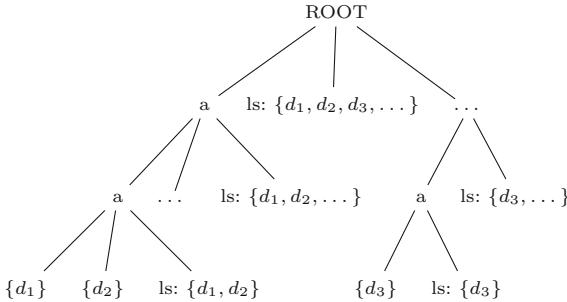


Fig. 6. GeoTree structure with list nodes

3.4 Retrieval of the Subtree Data

Given any geohash, we can query the tree for a set of nearby neighbouring geohashes by traversing down the GeoTree along some substring of that geohash. A longer length substring will correspond to a smaller radius in which neighbours will be returned. When the desired level is reached, the cached list node at that level can be queried for instant retrieval of the set of approximate k -NN of the geohash in question.

As a result of this structure's design, the GeoTree does not produce a distance measure for the items in the GeoTree. Rather, it clusters groups of nearby data points. While this does not allow for fine tuning of the search radius, it allows a set of data points which are geospatially close to the specified geohash to be retrieved in constant time.

3.5 Memory Requirement of the Data Structure

As each geohash is associated with only one character at each level of the GeoTree, only one node on each level will hold that geohash's data entry in its list node. Thus, each data entry is inserted into one single list node at every level of the tree. Given a tree of height h , this means that the data will be stored

in h different list nodes in addition to the one leaf node which the data receives. If the dataset is of size n , then there will be $(h + 1) * n$ data entries stored in the tree. However, as the height of the tree is fixed and specified prior to the building of the tree, the overall memory requirement of the GeoTree is $O(n)$. This can be further improved to only n data entries stored by collecting a set of the data once in memory and filling the list nodes with a list of pointers to the data entries, if necessary.

3.6 Technical Implementation

To touch briefly on the implementation of GeoTree [16], a nested hash map structure is used in order to store the tree. The root node is the root hash map of the nest, with the hash keys at this level corresponding to the letters of the level one nodes. Each of these keys point to a value which is another hash map containing keys corresponding to the level two letters of geohashes which have matching first letters with the parent key. The nesting process continues down to the leaf nodes (or terminal hash values in this case) in the same fashion described in Subsect. 3.1. The final hash key (representing the last character of the geohash) points to the list of data entries associated with that geohash.

3.7 Time Complexity

Building (Insertion). As hash maps offer $O(1)$ insertion, insertion of data at each level of the GeoTree is $O(1)$. Furthermore, due to the height of the tree, h , being constant and fixed, insertion of entries to the GeoTree is an $O(1)$ operation overall.

SCB Lookup. The $O(1)$ lookup of hash maps also means that the tree can be traversed in steps of $O(1)$ time. As the *list* nodes hold the SCB of every geohash substring possible from those in the dataset, and a maximum of h SCBs will need to be queried, it follows that any SCB lookup is also $O(1)$.

3.8 Comparison with the Prefix Tree (Trie)

The GeoTree data structure shares a number of similarities with the prefix tree or *trie* data structure [5]. A trie is a search tree which utilises its ordering and structure to increase searching efficiency across its inserted strings. Each branch represents a character and thus as you traverse down the trie, you build the prefix of a word, working toward an entire word at each leaf node.

This is very similar to the GeoTree, as the geohash encodings of properties take the place of words in this use case and traversing the GeoTree builds prefixes of geohash strings. Both data structures make use of structure to make search more efficient, however, in the case of the GeoTree, the ordering has geographical significance rather than the semantic meaning in the prefix tree.

One key difference between tries and GeoTrees lies in the subtree data caching step. As the GeoTree relies on being able to query every entry in the subtree of a particular node, the caching is necessary to quickly return a large number of property records. In the case of a prefix tree, it would be necessary to enumerate every path in the subtree to retrieve all of the words. In the use case which the GeoTree is being applied to, this would result in a significant increase in execution time over a very large dataset.

The GeoTree data structure could be thought of as a variant or augmentation of the trie, one which is specifically designed to give a fast, approximate solution to k -NN on geospatial datasets.

4 Real-World Performance

4.1 Application: House Price Index Algorithm

In order to test the performance of GeoTree in practice, we applied it to the computation of an Irish house price index. House price indexes and forecasting models have come under increased attention from a data mining context, with a view to improve the current methods of calculating and forecasting property price changes. Such algorithms could help identify price bubbles, facilitating preemptive measures to avoid another market collapse [6, 10, 11].

Many of these algorithms are based around the mix-adjusted median or central price tendency model, which requires a geospatial k -NN search [7, 14]. This approach is based on the principle that large amounts of aggregated data will cancel noise and result in a stable, smooth signal. It also offers the benefit of being less complex than the highly-theoretical hedonic regression model. It also requires less data than the repeat-sales model, in the sense of both quantity and time period spread [7, 14, 22].

Maguire et al. [14] introduced an enhanced central-price tendency model which outperformed the robustness of the hedonic regression method used by the Irish Central Statistics Office [21]. The primary limitation of this method is the algorithmic complexity and brute-force nature of the geospatial search, which impinges on its scalability to larger datasets, and restricts the introduction of further parameters. Our aim was to apply the GeoTree data structure to improve the execution time, scalability and robustness of this method. We re-implemented the algorithm used by [14] (described below), running the algorithm on the same data set (Irish Property Price Register) used in the original article as a control test for performance before introducing the GeoTree. For the purposes of algorithmic complexity calculation, we let n be the average number of house sales present in one month of the dataset, and let t be the number of months of data in the dataset.

Stage two (voting) of the original algorithm is executed as follows:

- ⇒ Iterate over each month, m , of the dataset
(t operations)

- ⇒ Iterate over each house, h , sold during m
(n operations)
 - ⇒ Iterate over houses sold in m to find the nearest to h (n operations*)

Stage four (stratification) of the **original** algorithm is executed as follows:

- ⇒ Iterate over each month, m , of the dataset
(t operations)
 - ⇒ Iterate over each house, h , sold during m
(n operations)
 - ⇒ Iterate over each month prior to m , m_p
($\frac{t-1}{2}$ operations)
 - ⇒ Iterate over houses sold in m_p to find the nearest to h (n operations*)

By introducing the GeoTree to the algorithm, the steps which formerly required an $O(n)$ iteration over all houses in the dataset to identify the nearest house (marked by an asterisk) now become an $O(1)$ GeoTree ranged proximity search operation. There is, however, a mild trade-off. Rather than returning the closest property to the house in question, the GeoTree structure instead returns everything in a small area around the house (formally, it returns the maximal length non-empty SCB for that house's geohash). The bucket can then be iterated over to find the true closest property, or an alternative strategy can be employed, such as taking the median price of all houses within the small area.

4.2 Performance Results

Table 1 compares the performance of the algorithms described previously with and without GeoTrees (on a database of 279,474 property sale records), including both single threaded execution time and multi-threaded execution time (running eight threads across eight CPU cores) on our test machine. The results using the GeoTree are marked with a + symbol.

4.3 Correlation

Despite the algorithmic alteration of taking the median price of a group of geo-hashed nearest neighbours, as opposed to the nearest neighbour per se, the house price indexes produced by the original algorithm and the GeoTree-enhanced version are very similar. Figure 7 shows both versions of the Residential Property Price Index (RPPI) superimposed. The two different versions yielded highly correlated outputs (Pearson's $r = 0.999$, Spearman's $\epsilon = 0.997$, Kendall's $\tau = 0.966$), revealing that GeoTree succeeded in delivering an almost identical index to the original one, though with major performance gains in execution time.

Table 1. Complexity and performance of the algorithms

Algorithm	Complexity	μ (1 core) ^a	β^b	μ (8 cores) ^a	β^b
Voting	$O(n^2t)$	233.54 s ^c	2.37%	46.73 s ^c	1.69%
Voting⁺	$O(nt)$	12.78 s ^c	1.68%	3.02 s ^c	0.69%
Stratify	$O\left(\frac{n^2t(t-1)}{2}\right)$	29.03 h	2.41%	4.19 h	1.89%
Stratify⁺	$O\left(\frac{nt(t-1)}{2}\right)$	$\in 0.05$ h (163.89 s)	1.71%	$\in 0.01$ h (39.63 s)	0.85%
Overall	$O\left(\frac{n^2t(t+1)}{2}\right)$	29.11 h	2.43%	4.21 h	1.90%
Overall⁺	$O\left(\frac{nt(t+1)}{2}\right)$	$\in 0.05$ h (177.73 s)	1.67%	$\in 0.01$ h (43.71 s)	0.79%

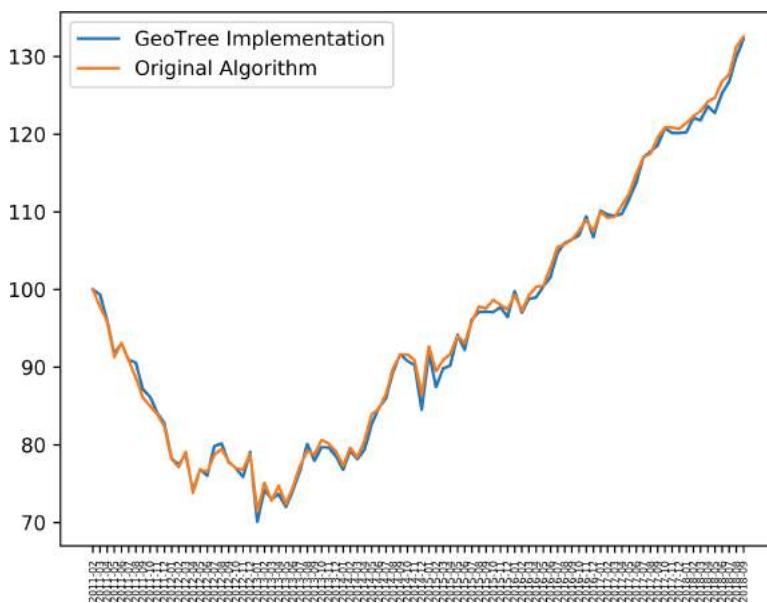
^a Execution times reported are the mean (μ) of ten trials.

^b Standard deviation (β) reported as a percentage of the mean (μ).

^c Includes build time for the dataset array/GeoTree on the dataset, as applicable.

^d All algorithms computed using an AMD Ryzen 2700X CPU.

^e All algorithms executed on the Irish Residential Property Price Register database of **279,474 property sale records** as of time of execution.

**Fig. 7.** Irish RPPI (GeoTree vs original), from 02-2011 to 09-2018

4.4 Scalability Testing

In order to test the scalability of the GeoTree, we obtained a dataset comprising 2,857,669 property sale records for California, and evaluated both the build and query time of the data structure. Table 2 shows mean build time and mean

query time on both 10% ($\sim 285,000$ records) and 100% (~ 2.85 million records) of the dataset. In this context, query time refers to the total time to perform **100 sequential queries**, as a single query was too fast to accurately measure.

The results demonstrate that the height of the tree has a modest effect on the build time, while dataset size has a linear effect on build time, thus supporting the claimed $O(n)$ build time with $O(1)$ insertion. Furthermore, query time is shown to remain constant regardless of both tree height and dataset size, with negligible differences in all instances.

Table 2. Scalability performance of GeoTree

Height h	4	5	6	7	8
Build Time (10%) ^a	17.63s (0.08 s)	18.10 s (0.10 s)	18.46 s (0.22 s)	18.84 s (0.08 s)	19.39 s (0.09 s)
Build Time (100%) ^b	179.67 s (0.58 s)	183.80 s (0.57 s)	183.99 s (0.52 s)	192.06 s (0.60 s)	194.31 s (0.94 s)
Query Time (10%) ^c	5.1 ms (0.3 ms)	5.2 ms (0.4 ms)	5.3 ms (0.9 ms)	5.3 ms (0.4 ms)	5.3 ms (0.5 ms)
Query Time (100%) ^c	5.4 ms (1.0 ms)	5.3 ms (0.9 ms)	5.5 ms (1.0 ms)	5.7 ms (1.3 ms)	5.6 ms (1.2 ms)

^a Build Time (10%) is the total time to insert 10% of dataset ($\in 285,000$ records)

^b Build Time (100%) is the total time to insert 100% of dataset ($\in 2.85\text{m}$ records)

^c Query Time consists of total time to execute 100 sequential neighbour queries on 10% and 100% of the dataset respectively

^d Times reported are in the format $\mu(\beta)$ calculated over ten trials

4.5 Discussion of Results

The results show that the GeoTree data structure offers the necessary scalability and speed of execution to expand to much larger geospatial datasets, including larger property price datasets. The biggest limitation of the GeoTree lies in the geospatial search distance ranges being linked to the length of the geohash string encoding, thus not being alterable to any desired distance. As a result, the algorithm loses a small amount of accuracy in comparison with the original, as discussed in Subsect. 4.3. Despite this, the substantial gains in execution time shown in Table 1 combined with the scalability offered by an $O(1)$ search algorithm demonstrated in Table 2 make a compelling case for a worthwhile trade-off in certain applications, where execution time would become too long with exact methods, such as in the property price index application shown.

Further improvements to the algorithm which could be explored in future research include querying just the surrounding squares of a geohash grid through a GeoTree search, rather than moving up an entire level. For example, in Fig. 1, a search for neighbours in *wx4g0c* which fails could explore neighbour *wx4g0f*

and the other adjacent neighbouring squares before falling back to searching through the entirety of $wx4g0$. This would likely restore some of the lost accuracy previously mentioned without introducing a large execution time overhead, should a mapping of the lettering patterns be computed beforehand and used in neighbour exploration.

5 Conclusion

We have shown that the GeoTree data structure introduced in this article offers an efficient $O(1)$ method for geospatial approximate k -NN search over a collection of geohashes. The application to a real-world property price index algorithm revealed significant reductions in execution time, and potentially opens the door for a real-time property price index. The data structure also performed well when applied to a much larger dataset, demonstrating its scalability. In conclusion, any data science problem which requires geospatial sampling around a particular area can employ the GeoTree for $O(1)$ retrieval of approximate neighbours, potentially enabling, for example, fast retrieval of locations of interest to map users, or geo-targeted advertisement and social networking updates.

References

1. Apostolico, A., Crochemore, M., Farach-Colton, M., Galil, Z., Muthukrishnan, S.: 40 years of suffix trees. *Commun. ACM* **59**(4), 66–73 (2016)
2. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Angela, Y.W.: An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM* **45**(6), 891–923 (1998)
3. Bao, J., Chow, C., Mokbel, M.F., Ku, W.: Efficient evaluation of k-range nearest neighbor queries in road networks. In: 2010 Eleventh International Conference on Mobile Data Management, pp. 115–124, May 2010
4. Beygelzimer, A., Kakade, S., Langford, J.: Cover trees for nearest neighbor. In: Proceedings of the 23rd International Conference on Machine Learning, ICML 2006, pp. 97–104. ACM, New York (2006)
5. De La Briandais, R.: File searching using variable length keys. In: Papers Presented at the the 3–5 March 1959, Western Joint Computer Conference, IRE-AIEE-ACM 1959 (Western), pp. 295–298. Association for Computing Machinery, New York (1959)
6. Diewert, W.E., de Haan, J., Hendriks, R.: Hedonic regressions and the decomposition of a house price index into land and structure components. *Econometr. Rev.* **34**(1–2), 106–126 (2015)
7. Goh, Y.M., Costello, G., Schwann, G.: Accuracy and robustness of house price index methods. *Hous. Stud.* **27**(5), 643–666 (2012)
8. Guttman, A.: R-trees: a dynamic index structure for spatial searching. *SIGMOD Rec.* **14**(2), 47–57 (1984)
9. Hand, D.J.: Data mining based in part on the article “data mining” by David Hand, which appeared in the encyclopedia of environmetrics. American Cancer Society (2013)

10. Jadevicius, A., Huston, S.: Arima modelling of Lithuanian house price index. *Int. J. Hous. Mark. Anal.* **8**(1), 135–147 (2015)
11. Klotz, P., Lin, T.C., Hsu, S.-H.: Modeling property bubble dynamics in Greece, Ireland, Portugal and Spain. *J. Eur. Real Estate Res.* **9**(1), 52–75 (2016)
12. Kothuri, R.K.V., Ravada, S., Abugov, D.: Quadtree and r-tree indexes in oracle spatial: a comparison using GIS data. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 546–557. ACM (2002)
13. Lee, J.-G., Kang, M.: Geospatial big data: Challenges and opportunities. *Big Data Res.* **2**(2), 74–81 (2015). Visions on Big Data
14. Maguire, P., Miller, R., Moser, P., Maguire, R.: A robust house price index using sparse and frugal data. *J. Prop. Res.* **33**(4), 293–308 (2016)
15. McGinnis, W.: Pygeohash (2017). [Python]
16. Miller, R.: Geotree data structure code implementation (2020). <https://github.com/robertmiller72/GeoTree/blob/master/GeoTree.py>. [Python]
17. Moussalli, R., Srivatsa, M., Asaad, S.: Fast and flexible conversion of geohash codes to and from latitude/longitude coordinates. In: 2015 IEEE 23rd Annual International Symposium on Field-Programmable Custom Computing Machines, pp. 179–186, May 2015
18. Moussalli, R., Asaad, S.W., Srivatsa, M.: Enhanced conversion between geohash codes and corresponding longitude/latitude coordinates (2015)
19. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(11), 2227–2240 (2014)
20. Niemeyer, G.: geohash.org is public! (2008). Accessed 02 May 2019
21. O'Hanlon, N.: Constructing a national house price index for Ireland. *J. Stat. Soc. Inq. Soc. Ireland* **40**, 167–196 (2011)
22. Prasad, N., Richards, A.: Improving median housing price indexes through stratification. *J. Real Estate Res.* **30**(1), 45–72 (2008)
23. Robusto, C.C.: The cosine-haversine formula. *Am. Math. Mon.* **64**(1), 38–40 (1957)
24. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. *SIGMOD Rec.* **24**(2), 71–79 (1995)
25. Safar, M.: K nearest neighbor search in navigation systems. *Mob. Inf. Syst.* **1**(3), 207–224 (2005)



Prediction Interval of Future Waiting Times of the Two-Parameter Exponential Distribution Under Multiply Type II Censoring

Shu-Fei Wu^(✉)

Department of Statistics, Tamkang University, New Taipei, Taiwan, ROC
100665@mail.tku.edu.tw

Abstract. We use the general weighted moments estimator (GWME) of the scale parameter of the two-parameter exponential distribution based on a multiply type II censored sample to construct the pivotal quantities for the use of the prediction intervals of future waiting times. This estimator has been shown to have better performance than the other fourteen estimators in terms of mean square error. At last, one real life example is given to illustrate the prediction intervals based on GWMEs.

Keywords: Type II multiply censored sample · Exponential distribution · General weighted moments estimator · Prediction interval

1 Introduction

In most literatures of reliability, the exponential distribution is widely used as a model of lifetime data. There are many applications of exponential distribution in the analysis of reliability and the life test experiments. See for example, Johnson et al. (1994) [4]. The failure time Y follows a two-parameter exponential distribution if the probability density function (p.d.f.) of Y is given by $f(y) = \frac{1}{\theta} \exp\left(-\frac{y-\mu}{\theta}\right)$, $y \geq 0$, $\mu > 0$, $\theta > 0$, where μ is the location parameter and θ is the scale parameter. The location parameter of two-parameter exponential distributions are so-called threshold values or "guaranteed time" parameters in reliability and engineering. In dose-response experiments, this distribution is generally used to model the effective duration of a drug, where the location parameter μ is regarded as the guaranteed effective duration and the scale parameter θ is referred as the mean effective duration in addition to μ .

In life testing experiments, the experimenters may not be able to obtain the lifetimes of all items that are put on test due to the artificial mistakes or for implementing some purposes of experimental designs. Suppose that there are n items are put on the life test and the first r , middle l and the lasts are unobserved or missing, this type of censoring is called the type II multiply censoring. Wu et al. (2011) proposed the simultaneous confidence intervals for all distances from the extreme populations for two-parameter exponential populations based on the multiply type II censored samples. Wu (2016) proposed the prediction interval for the future waiting times for one-parameter exponential

distribution based on type II multiply censored sample. For two-parameter exponential distribution, Wu (2015) [8] proposed some general WMEs (GWMEs) by assigning a single weight to each observation instead of only considering two weights in Wu and Yang (2002) [6] for exponential distribution under multiply type II censoring. The simulation comparison results show that the GWMEs outperforms the 12 weighted moments estimators proposed by Wu and Yang (2002) and approximate maximum likelihood estimator (AMLE) by Balakrishnan (1990) [2] and the best linear unbiased estimator (BLUE) by Balasubramanian and Balakrishnan (1992) [3] in terms of the exact mean squared errors (MSEs) in most cases for exponential distribution. Since GWMEs perform better than other 14 methods. Wu (2019) [7] made use of GWMEs to build the prediction intervals for the future observations. Some users may be interested in the prediction of future waiting times between two failure times. For this kind of problem, we utilized GWMEs to construct a pivotal quantity and build the prediction interval of future waiting times based on the corresponding pivotal quantity. The structure of this research is organized as follows: In Sect. 2, the general WME for the two-parameter exponential distribution is introduced. In Sect. 3, the prediction intervals of future waiting times are proposed. The percentiles of the pivotal quantities are also tabulated for practical use in this section. One real life example to illustrate the proposed intervals is given in Sect. 4. At last, some discussions are given in Sect. 5.

2 The General Weighted Moments Estimation of the Scale Parameter of the One-Parameter Exponential Distribution

Suppose that the lifetimes Y follows a two-parameter exponential distribution with p.d.f. given by $f(y) = \frac{1}{\theta} \exp\left(-\frac{y-\mu}{\theta}\right)$, $y \geq 0$, $\mu > 0$, $\theta > 0$, where μ is the location parameter and θ is the scale parameter. Let $Y_{(r+1)} < \dots < Y_{(r+k)} < Y_{(r+k+l+1)} < \dots < Y_{(n-s)}$ be the available type II multiply censored sample from the above distribution.

Let $Y_{(i)}^* = Y_{(i)} - Y_{(r+1)}$, $i = r+2, \dots, r+k, r+k+l+1, \dots, n-s$. The general WME for the scale parameter is defined as $\tilde{\theta}^* = W_{r+2}^* Y_{(r+2)}^* + \dots + W_{r+k}^* Y_{(r+k)}^* + W_{r+k+l+1}^* Y_{(r+k+l+1)}^* + \dots + W_{n-s}^* Y_{(n-s)}^*$, where $W_{r+2}^*, \dots, W_{r+k}^*, \dots, W_{r+k+l+1}^*, \dots, W_{n-s}^*$ are $n-r-l-1$ weights assigned to $Y_{(r+2)}^*, \dots, Y_{(r+k)}^*, \dots, Y_{(r+k+l+1)}^*, \dots, Y_{(n-s)}^*$. Let $\tilde{Y}^* = [Y_{(r+2)}^*, \dots, Y_{(r+k)}^*, \dots, Y_{(r+k+l+1)}^*, \dots, Y_{(n-s)}^*]^t$ and $\tilde{W}^* = [W_{r+2}^* \dots W_{r+k}^* \dots W_{r+k+l+1}^* \dots W_{n-s}^*]$, where $[X]^t$ represents the transpose of a vector X . Then the GWME for the scale parameter θ can be rewritten as $\tilde{\theta}^* = \tilde{W}^* \tilde{Y}^*$.

The weight vector $\tilde{W}^* = [W_{r+2}^* \dots W_{r+k}^* \dots W_{r+k+l+1}^* \dots W_{n-s}^*]$ is determined to minimize the mean square error (MSE) of the proposed GWME. From Wu (2015), the optimal weight vector is determined as $\tilde{W}^* = A^{*-1} \tilde{a}^*$, where

$$A^* = \begin{bmatrix} b_{r+2,r+2}^* + a_{r+2}^{*2} & b_{r+2,r+3}^* + a_{r+2}^* a_{r+3}^* & \dots & b_{r+2,n-s}^* + a_{r+2}^* a_{n-s}^* \\ b_{r+2,r+3}^* + a_{r+2}^* a_{r+3}^* & b_{r+3,r+3}^* + a_{r+3}^{*2} & \dots & b_{r+3,n-s}^* + a_{r+3}^* a_{n-s}^* \\ \vdots & \ddots & \vdots & \vdots \\ b_{r+2,n-s}^* + a_{r+2}^* a_{n-s}^* & \dots & \dots & b_{n-s,n-s}^* + a_{n-s}^{*2} \end{bmatrix},$$

$a^* = (a_{r+2}^*, \dots, a_{r+k}^*, a_{r+k+l+1}^*, \dots, a_{n-s}^*)$ is the mean vector of the random vector \underline{Y}^* and $B^* = [b_{i,j}^*]_{i=r+2, \dots, r+k, r+k+l+1, \dots, n-s; j=r+2, \dots, r+k, r+k+l+1, \dots, n-s}$ is the covariance matrix of the random vector \underline{Y}^* . The GWME with minimum MSE is obtained as

$$\tilde{\theta}^* = \underline{W}^{*T} \underline{Y}^* = A^{*-1} \underline{a}^* \underline{Y}^* \quad (1)$$

The minimum MSE of $\tilde{\theta}^*$ is

$$\text{MSE}(\tilde{\theta}^*) = \left(\underline{W}^{*T} B^* \underline{W}^* + (\underline{W}^{*T} a^* - 1)^2 \right) \theta^2 \quad (2)$$

3 Intervals of Future Waiting Time

In order to predict the waiting Times, the pivotal quantity is considered as $V = (Y_{(j)} - Y_{(j-1)}) / \tilde{\theta}^*$, $n - s < j \leq n$ based on the GWME $\tilde{\theta}^*$ defined in (1). Since $\frac{Y_{(1)}}{\theta}, \dots, \frac{Y_{(n)}}{\theta}$ are the n order statistics from a standard exponential distribution and $\frac{\tilde{\theta}^*}{\theta}, \dots, \frac{\underline{W}^{*T} \underline{Y}^*}{\theta}$ is a linear combination of n order statistics from a standard exponential distribution, the distribution of pivotal quantity $V = \left(\frac{Y_{(j)} - Y_{(j-1)}}{\theta} \right) / \frac{\tilde{\theta}^*}{\theta}$ is independent of θ , $n - s < j \leq n$. Let $V(\delta; n, j, r, k, l, s)$ be the δ percentile of the distribution of V satisfying $P(V \leq V(\delta; n, j, r, k, l, s)) = \delta$.

Making use of the pivotal quantity V , the prediction interval of waiting time $Y_{(j)} - Y_{(j-1)}$, $n - s < j \leq n$ is proposed in the following Theorem.

Theorem 1: For type II multiply censored sample $Y_{(r+1)} < \dots < Y_{(r+k)} < Y_{(r+k+l+1)} < \dots < Y_{(n-s)}$, the prediction interval of future waiting times $Y_{(j)} - Y_{(j-1)}$, $n - s < j \leq n$ is $(V(\frac{\alpha}{2}; n, j, r, k, l, s)\tilde{\theta}^*, V(1 - \frac{\alpha}{2}; n, j, r, k, l, s)\tilde{\theta}^*)$

Proof: Observed that

$$\begin{aligned} 1 - \alpha &= P\left(V\left(\frac{\alpha}{2}; n, j, r, k, l, s\right) \leq \left(\frac{Y_{(j)} - Y_{(j-1)}}{\tilde{\theta}^*}\right) \leq V\left(1 - \frac{\alpha}{2}; n, j, r, k, l, s\right)\hat{\theta}\right) \\ &= P\left(V\left(\frac{\alpha}{2}; n, j, r, k, l, s\right)\tilde{\theta}^* \leq Y_{(j)} - Y_{(j-1)} \leq V\left(1 - \frac{\alpha}{2}; n, j, r, k, l, s\right)\tilde{\theta}^*\right) \quad \square \end{aligned}$$

Since the exact distributions of V is too hard to derive algebraically, the δ percentile of the distribution of V is obtained based on Monte Carlo simulation. Moreover, all the simulations were run with the aid of AbSoft Fortran Inclusive of IMSL (1999) [1]. In the simulation, 600,000 replicates are used to compute the percentiles of V for each combination of n, r, k, l, s, j , where $j = n - s + 1, \dots, n$. Due to the limitation of the number of pages, only part of the percentiles of V are given in Table 1, for $\delta = 0.005, 0.010, 0.025, 0.050, 0.100, 0.900, 0.950, 0.975, 0.990, 0.995$ under $n = 12, 24, 36$ (see Table 1). Any specific percentile $V(\delta; n, j, r, k, l, s)$ for any censoring scheme (n, r, k, l, s) for the j th waiting time between the j th future observation and the previous one, $j = n - s + 1, \dots, n$, can be obtained by the software program provided by author.

Table 1. The δ percentile of the pivotal quantity $V = (Y_{(l)} - Y_{(n-s)}) / \tilde{\theta}^*$ and $P(V \leq V(\delta; n, r, k, l, s)) = \delta$

δ															
n	r	k	l	s	j	0.005	0.010	0.025	0.050	0.100	0.900	0.950	0.975	0.990	0.995
12	2	5	2	2	11	0.0029	0.0058	0.0145	0.0295	0.0610	1.5659	2.1491	2.7855	3.7410	4.5550
12	2	5	2	2	12	0.0057	0.0115	0.0292	0.0593	0.1223	3.1302	4.2905	5.5651	7.4760	9.1210
12	3	5	2	1	12	0.0057	0.0115	0.0289	0.0590	0.1221	3.1466	4.3155	5.6153	7.5330	9.1530
12	3	5	1	2	11	0.0029	0.0058	0.0147	0.0300	0.0620	1.6406	2.2699	2.9785	4.0540	4.9780
12	3	5	1	2	12	0.0058	0.0117	0.0298	0.0605	0.1247	3.2833	4.5408	5.9647	8.1440	9.9840
12	2	5	3	1	12	0.0058	0.0115	0.0289	0.0584	0.1204	3.0439	4.1545	5.3660	7.1270	8.6290
12	2	5	1	3	10	0.0020	0.0039	0.0099	0.0201	0.0413	1.0935	1.5169	1.9903	2.7050	3.3240
12	2	5	1	3	11	0.0029	0.0058	0.0148	0.0300	0.0617	1.6446	2.2701	2.9771	4.0680	4.9940
12	2	5	1	3	12	0.0058	0.0119	0.0298	0.0606	0.1248	3.2821	4.5493	5.9623	8.1300	9.9930
12	1	5	2	11	0.0028	0.0057	0.0144	0.0292	0.0598	1.5122	2.0578	2.6606	3.5490	4.2780	
12	1	5	3	2	12	0.0057	0.0114	0.0286	0.0582	0.1200	3.0277	4.1241	5.3219	7.0760	8.5480
12	1	5	2	3	10	0.0019	0.0038	0.0096	0.0196	0.0406	1.0409	1.4314	1.8602	2.5050	3.0490
12	1	5	2	3	11	0.0029	0.0058	0.0146	0.0296	0.0608	1.5678	2.1462	2.7957	3.7460	4.5530
12	1	5	2	3	12	0.0058	0.0115	0.0289	0.0587	0.1209	3.1237	4.2827	5.5682	7.4750	9.0610
12	4	5	1	12	0.0059	0.0118	0.0296	0.0601	0.1241	3.2901	4.5516	5.9824	8.1700	10.0450	
12	1	5	4	1	12	0.0056	0.0112	0.0283	0.0577	0.1192	2.9602	4.0191	5.1518	6.8360	8.2240
24	2	5	2	2	23	0.0026	0.0053	0.0134	0.0270	0.0557	1.2897	1.7100	2.1431	2.7470	3.2260

(continued)

Table 1. (continued)

														δ	
24	2	5	2	2	24	0.0054	0.0107	0.0268	0.0542	0.1109	2.5777	3.4134	4.2800	5.4770	6.4320
24	3	5	2	1	24	0.0053	0.0105	0.0269	0.0543	0.1113	2.5807	3.4173	4.2908	5.4810	6.4340
24	3	5	1	2	23	0.0026	0.0053	0.0134	0.0271	0.0556	1.2948	1.7212	2.1632	2.7830	3.2660
24	3	5	1	2	24	0.0054	0.0107	0.0268	0.0541	0.1115	2.5945	3.4411	4.3191	5.5380	6.4740
24	2	5	3	1	24	0.0052	0.0105	0.0265	0.0537	0.1108	2.5683	3.3970	4.2557	5.4460	6.4010
24	2	5	1	3	22	0.0018	0.0036	0.0089	0.0181	0.0372	0.8633	1.1435	1.4372	1.8410	2.1580
24	2	5	1	3	23	0.0027	0.0053	0.0133	0.0270	0.0555	1.2997	1.7244	2.1710	2.7740	3.2560
24	2	5	1	3	24	0.0053	0.0107	0.0270	0.0545	0.1117	2.5971	3.4364	4.3216	5.5440	6.5200
24	1	5	3	2	23	0.0026	0.0052	0.0133	0.0270	0.0554	1.2791	1.6925	2.1298	2.7180	3.1730
24	1	5	3	2	24	0.0051	0.0104	0.0265	0.0541	0.1108	2.5608	3.3924	4.2408	5.4300	6.3830
24	1	5	2	3	22	0.0018	0.0035	0.0089	0.0180	0.0370	0.8602	1.1420	1.4348	1.8320	2.1440
24	1	5	2	3	23	0.0027	0.0053	0.0133	0.0270	0.0555	1.2853	1.7038	2.1389	2.7440	3.2140
24	1	5	2	3	24	0.0053	0.0106	0.0266	0.0539	0.1113	2.5809	3.4211	4.2956	5.4800	6.4340
24	4	5	1	1	24	0.0053	0.0108	0.0266	0.0542	0.1118	2.5996	3.4482	4.3324	5.5620	6.5150
24	1	5	4	1	24	0.0055	0.0107	0.0265	0.0536	0.1105	2.5513	3.3780	4.2261	5.3800	6.2980
36	2	5	2	2	35	0.0025	0.0051	0.0130	0.0264	0.0543	1.2380	1.6277	2.0220	2.5640	2.9710
36	2	5	2	2	36	0.0052	0.0104	0.0262	0.0532	0.1095	2.4630	3.2430	4.0444	5.1280	5.9740
36	3	5	2	1	36	0.0051	0.0104	0.0262	0.0530	0.1092	2.4768	3.2551	4.0546	5.1290	5.9660

(continued)

Table 1. (*continued*)

4 Example

In this section, we use the example of times to breakdown of an insulating fluid between electrodes recorded at five different voltages (Nelson (1982 p. 252)) [5] to demonstrate the prediction interval of future waiting times. Such a distribution of time to breakdown is usually assumed to be exponential distributed in engineering theory. We choose 35 kV, and the multiple type II censored data with $n = 12$, $r = 2$, $k = 3$, $l = 1$ and $s = 5$ is: $\text{---}, \text{---}, 41, 87, 93, \text{---}, 116, \text{---}, \text{---}, \text{---}, \text{---}$. The weights are 0.23568, 0.12544, 0.19776, 0.8058 and the estimated scale parameter is

$$\tilde{\theta}^* = W_{r+2}^* Y_{(r+2)}^* + \dots + W_{r+k}^* Y_{(r+k)}^* + W_{r+k+l+1}^* Y_{(r+k+l+1)}^* + \dots + W_{n-s}^* Y_{(n-s)}^*$$

$$W_4^* (Y_{(4)}^* - Y_{(3)}^*) + W_5^* (Y_{(5)}^* - Y_{(3)}^*) + 7_5^* (Y_{(7)}^* - Y_{(3)}^*) = 0.20047 * 46 + 0.31604 * 52 + 1.28774 * 75 = 1.22.2362$$

Using Theorem 1, the 90% and 95% prediction intervals for $Y_{(8)} - Y_{(7)}$, $Y_{(9)} - Y_{(8)}$, $Y_{(10)} - Y_{(9)}$, $Y_{(11)} - Y_{(10)}$, $Y_{(12)} - Y_{(11)}$ are obtained in Table 2.

Table 2. 90% and 95% prediction interval for future waiting times times $Y_{(j)} - Y_{(j-1)}$, $j = 8, \dots, 12$.

90%		
Waiting time	$U(0.05; 12, j, 2, 3, 1, 5)$, $U(0.95; 12, j, 2, 3, 1, 5)$	Prediction interval
$Y_{(8)} - Y_{(7)}$	0.0129, 1.1147	(1.576847, 136.256692)
$Y_{(9)} - Y_{(8)}$	0.0161, 1.3932	(1.968003, 170.299474)
$Y_{(10)} - Y_{(9)}$	0.0216, 1.8603	(2.640302, 227.396003)
$Y_{(11)} - Y_{(10)}$	0.0323, 2.7938	(3.948229, 341.503496)
$Y_{(12)} - Y_{(11)}$	0.0645, 5.5884	(7.884235, 683.104780)
95%		
Waiting time	$U(0.025; 12, j, 2, 3, 1, 5)$, $U(0.975; 12, j, 2, 3, 1, 5)$	Prediction interval
$Y_{(8)} - Y_{(7)}$	0.0064, 1.5165	(0.7823117, 185.3711973)
$Y_{(9)} - Y_{(8)}$	0.0079, 1.8969	(0.965666, 231.869848)
$Y_{(10)} - Y_{(9)}$	0.0106, 2.5295	(1.295704, 309.196468)
$Y_{(11)} - Y_{(10)}$	0.0157, 3.7951	(1.919108, 463.898603)
$Y_{(12)} - Y_{(11)}$	0.0317, 7.6069	(3.874888, 929.838550)

5 Discussion

In this paper, the GWMEs are used to construct a pivotal quantity to build a prediction interval for future waiting times for two-parameter exponential distribution. The percentiles of proposed pivotal quantity are obtained by Monte-Carlo method and tabulated in Table 1. Theorem 1 is proposed to build the prediction intervals based on GWME for type II multiply censored sample. At last, one real life example is given to illustrate the proposed prediction intervals of future waiting times for a specific censoring scheme. In the future, we should think of some other estimators to improve the performance of the prediction intervals. We can also investigate the prediction intervals for other lifetime distributions such as Pareto distribution.

Acknowledgment. The author's research was supported by Ministry of Science and Technology MOST 108-2118-M-032-001- and MOST 109-2118-M-032 -001 -MY2 in Taiwan, ROC.

References

1. AbSoft Fortran: (Inclusive of IMSL) 4.6, Copyright (c), Absoft Crop. (1999)
2. Balakrishnan, N.: On the maximum likelihood estimation of the location and scale parameters of exponential distribution based on multiply Type II censored samples. *J. Appl. Stat.* **17**, 55–61 (1990)
3. Balasubramanian, K., Balakrishnan, N.: Estimation for one- and two-parameter exponential distributions under multiple Type-II censoring. *Stat. Papers* **33**(1), 203–216 (1992). <https://doi.org/10.1007/BF02925325>
4. Johnson, N.L., Kotz, S., Balakrishnan, N.: Continuous Univariate Distributions, vol. 1. Wiley, Hoboken (1994)
5. Nelson, W.: Applied Life Data Analysis. Wiley, New York (1982)
6. Wu, J.W., Yang, C.C.: Weighted moments estimation of the scale parameter for the exponential distribution based on a multiply Type II censored sample. *Qual. Reliab. Eng. Int.* **18**, 149–154 (2002)
7. Wu, S.F.: Prediction interval of the future observations of the two-parameter exponential distribution under multiply Type II censoring. *ICIC Exp. Lett.* **13**(11), 1073–1077 (2019)
8. Wu, S.F.: The general weighted moment estimator of the scale parameter of the two-parameter exponential distribution under multiply type II censoring. *ICIC Exp. Lett. (ICIC-EL)* **9**(11), 3081–3085 (2015)



Predictive Models as Early Warning Systems: A Bayesian Classification Model to Identify At-Risk Students of Programming

Ashok Kumar Veerasamy¹(✉), Mikko-Jussi Laakso¹, Daryl D’Souza²,
and Tapio Salakoski¹

¹ University of Turku, Turku, Finland
askuve@utu.fi

² RMIT University, Melbourne, Australia

Abstract. The pursuit of a deeper understanding of factors that influence student performance outcomes has long been of interest to the computing education community. Among these include the development of effective predictive models to predict student academic performance. Predictive models may serve as early warning systems to identify students at risk of failing or quitting early. This paper presents a class of machine learning predictive models based on Naive Bayes classification, to predict student performance in introductory programming. The models use formative assessment tasks and self-reported cognitive features such as prior programming knowledge and problem-solving skills. Our analysis revealed that the use of just three variables was a good fit for the models employed. The models that used in-class assessment and cognitive features as predictors returned best at-risk prediction accuracies, compared with models that used take-home assessment and cognitive features as predictors. The prediction accuracy in identifying at-risk students on unknown data for the course was 71% (overall prediction accuracy) in compliance with the area under the curve (ROC) score (0.66). Based on these results we present a generic predictive model and its potential application as an early warning system for early identification of students at risk.

Keywords: Early warning systems · Formative assessment tasks · Predictive data mining models · Problem Solving Skills

1 Introduction

Programming is fundamental to computer science (CS) and cognate disciplines, and typically offered as a non-CS major. However, the difficulty of learning to program steers non-CS students away from programming courses, and to select alternative courses [1]. Many students fail or perform poorly in programming even as CS education has seen improvements in methods of teaching programming [2] with failure rates continuing to be the range 28–32% [3, 4]. Accordingly, the pursuit of a better understanding of factors that influence student performance outcomes has long been of interest, and includes the development of early-prediction models to predict academic performance, in turn to

identify potentially at-risk students [5–7]. However, the predictor variables and associated machine learning algorithms are typically influenced by contextual variations such as class size and academic settings [8, 9]. In addition, it is widely accepted that parsimony is important in model building [10, 11]. This paper proposes a genre of simple, parsimonious predictive model(s), which account for variations in academic setting, such as academic environment and student demography. The models also assume that concepts taught early in semester typically impact on understanding of latter concepts; hence analysing results of early formative assessments may provide opportunities to assess student-learning outcomes and to identify poorly motivated learners, early in the semester.

This study adopted the Naive Bayes classification for our proposed predictive model. K-fold cross-validation was used to evaluate model. An additional objective was to explore the relationship among the selected predictor variables and propose a genre of parsimonious predictive model(s) for incorporation in an early warning system (EWS), to identify at-risk students early and to facilitate appropriate interventions. Towards these objectives, the paper addresses the following research questions (RQs).

RQ1. Which measures provide the most value for predicting student performance: perceived problem-solving skills, prior programming knowledge, selected formative assessment results?

RQ2. How suitable is the Naive Bayes classification based model for incorporation in an early warning system, to identify students in need of early assistance?

The remainder of the paper is organized as follows. Related work (Sect. 2) surveys literature relevant to previous work. Research methodology (Sect. 3) describes the methods used to address our research questions. Data analysis and results (Sect. 4) presents our findings, which we discuss in depth in discussion (Sect. 5). Finally, conclusion (Sect. 6) summarizes our findings and presents limitations in terms of how well the foregoing research questions are answered; we also identify some related future work directions.

2 Related Work

This section highlights important past contributions of relevance to the work reported in this paper, including: predictors of academic performance; modelling of predictors; Naive Bayes classification and early warning systems.

2.1 Predictors of Student Performance

We limited our study to three variables for predicting student academic performance: problem-solving skills, prior programming knowledge and formative assessment performance. Problem solving is a metacognitive activity, which reveals the way a person learns and experiences different aspects of the problem-solving process [12]. It is considered as a basic skill for students in study and work [13, 14]. Student problem-solving skills can predict student study habits and academic performance [15]. Studies have

also revealed that there is a relationship between problem-solving proficiency and academic achievement [16, 17]. Marion et al. [18] noted that programming should not be considered as just a coding skill but as a way of thinking, decomposing and solving problems, implying that problem solving may impact student performance in programming courses. Research in the discipline of computer science has also highlighted that many students lack problem-solving skills [19, 20].

Another important variable is prior knowledge, which is defined as an individual's prior personal stock of information, skills, experiences, beliefs and memories. Prior knowledge is one of the most important factors that influence learning and student performance [21, 22]. Studies have been conducted on the impact of prior knowledge in programming courses, sometimes with mixed results [23–25]. Reviews of computing educational studies have found that both prior knowledge and problem-solving skills are required skills for CS students [24, 26].

Formative assessment is an effective instructional strategy to measure student-learning outcomes [27, 28]. Formative assessment tasks take place during the course of study and instructors often examine selected formative assessment task results to observe and assess student improvement. Students are aware that completing formative assessment tasks may lead to improved final grades [29]. For example, homework is a type of formative assessment task to test comprehension [30]. Educators use homework to identify where students are struggling, in order to assist them and to address their problems [31]. There have been several studies conducted on the impact of homework on student performance [31, 32]. Veerasamy et al. [31] reported that marks achieved in homework and class demonstrations have a significant positive impact on final examination results. Similarly, computer-based or online-tutorials have positive effects on student academic performance in economics, mathematics and science courses [33, 34]. Moreover, several models have included formative assessment scores to predict student performance [6, 7].

2.2 Predictive Data Mining Modelling for Student Performance in Computing Education

Predictive modelling employs classifiers or regressors to formulate a statistical model to forecast or predict future outcomes. Predictive models have been employed in several studies to automatically identify students in need of assistance in programming courses, based on early course work [7, 35–38]. For example, Porter et al. [36] used correlation coefficient and visualization techniques to predict student success at the end of term, examining clicker question performance data collected in peer instruction classrooms. Liao et al., [38, 39] conducted studies by using student clicker data as input, collected in a peer instruction pedagogy setting to identify at risk CS1 students. In a later study [7], they explored the value of different data sources as inputs to predict student performance in multiple courses; these inputs included the collected clicker data, take-home assignment and online quiz grades, and final grades obtained from prerequisite courses. They employed Linear regression, Support vector machine and Logistic regression machine learning algorithms in these studies respectively for prediction of student performance in computing education. However, the earlier study [39] did not provide sufficient detail why Linear regression was selected over other machine learning techniques. The use of

clickers and the peer instruction pedagogy raised concerns about whether this approach could be applied to courses that did not employ instrumented IDEs, or the peer instruction pedagogy as per later studies [7, 38]. In addition, they may not have had access to grades attained in the prerequisite courses for analysis [7]. The methodologies used in the aforementioned studies cannot be applied to online distance courses. Substantial student collaboration effort is required from students located at different geographical locations, and it may be challenging for instructors to obtain or access relevant data for predictive analysis. In addition, it is not yet clear which machine learning algorithms are preferable in this context.

2.3 Predictive Modelling with Naive Bayes Classification

Naive Bayes classification (NBC) is a supervised machine-learning algorithm for binary and multi-class classification problems. It is a simple statistical classifier and based on Bayes' probability theorem. NBC models considered as an effective choice for predicting student performance [40], and their use has demonstrated improved performance over other classification methods [41]. For example, Agrawal et al. [42] analysed various machine learning classification algorithms and inferred that NBC is effective for student final exam performance prediction. However, Bergin et al. found that although Naive Bayes had the highest prediction accuracy for predicting novice programming success, there were no significant statistical differences between the prediction accuracies of Naive Bayes and Logistic regression, Support vector machine, Artificial neural network and Decision tree classifiers, in predicting introductory programming student performance [43]. In addition, the study reported in this paper, we explored various other machine-learning techniques, such as Random forest, C5.0 and Support vector machine, for predicting student performance. It was identified that NBC provided better overall prediction and at-risk balanced accuracy across our datasets compared to other machine learning models. So, we deployed Naive Bayes classification for this study.

Most of the studies around predictive model development and validation used K-fold cross-validation to evaluate model performance [7, 40, 43]. Borra et al. [44] measured the prediction error of the model by employing estimators such as leave-one-out, parametric and non-parametric bootstrap, as well as cross-validation methods. They reported repeated K-fold cross-validation estimator and parametric bootstrap estimator outperformed the leave-one-out and hold-out estimators. As such, in this study, we developed the Naive Bayes based classification model and used K-fold- cross-validation for model evaluation.

2.4 Early Warning Systems (EWS) in Education

The term “early warning system” (EWS) is not new and has attracted attention to support instructors and students [45, 46]. The EWS acts as a student progress indicator, enabling educators to support students who perform progressively poorly, before they drop out; for example, Krumm et al. [46] designed a project called Student Explorer as a part of learning management system (LMS) for academic advising in undergraduate engineering courses. They examined the accumulated LMS data to identify students who

needed academic support and identified factors that influenced academic advisor decisions. Higher educational institutions are placing greater emphasis on improving student retention, performance and support, and hence they have begun to urge educators to use EWSs, such as Course Signals and Student Explorer, to implement effective pedagogical practices to improve student performance [46–48]. However, these EWSs have some shortcomings. First, academic advisors surmise that using EWSs in academic settings is time consuming. Second, many instructors have difficulties in integrating EWSs into their regular work practices as existing EWSs do not appear to be user-friendly. Third, studies also confirmed that identifying a reliable set of predictors to develop early warning systems/early prediction tools is a challenging task [49, 50]. Fourth, EWSs that have been designed for online courses or which rely heavily on learning management systems (LMS) data, and not on student cognitive and performance data, may not be suitable as data sources [46, 47]. Moreover, in programming, several learning activities take place outside of the LMS [51]. Hence, EWS tools and strategies developed on the basis of student cognitive and performance data, best serve the early identification of at-risk students, in order to support their timely learning [45].

In summary, this paper contributes the following: (i) Development of a simple model(s) with explanatory predictor variables selected on the basis of prior work; (ii) Adoption of the NBC algorithm with K-fold cross-validation, to develop predictive models and to explore the predictive capabilities of selected variables; (iii) Identification of models with reliable sets of predictors, which may be suitable in (academic) early warning systems, for courses that utilise assessment tasks and a final exam.

3 Research Methodology

The aim of this study was establish a predictive model to predict student final programming grades in introductory programming courses. This study used Naive Bayes classification based predictive modelling with the following inputs: student perceived problem-solving skills; prior knowledge in programming; and formative assessment in the form of homework and demo/tutorial exercise scores, for the first four weeks of semester. Student final exam grades represented the output of the model.

3.1 Description of the Course and Data Sources

The initial data collected for model development was derived from assessment activities of university students enrolled in two classroom-based courses: *Introduction to Programming* and *Algorithms and Programming*, both offered during the autumn semester of 2016. The dataset collected in the autumn semester of 2017 was then employed as the unknown data set to test the performance (for generalization) of our predictive models. *Introduction to Programming* is offered in English and *Algorithms and Programming* in Finnish, and both courses are designed for students with no prior knowledge in programming. The duration of the courses is 11 weeks and 8 weeks respectively, for *Introduction to Programming* and *Algorithms and Programming*. Both courses use ViLLE as the learning management system (LMS) to support technology-enhanced classes. ViLLE is mainly used for programming students, to deliver and manage course content,

such as lecture notes, formative and summative assessment tasks for programming students. Student academic data was collected via ViLLE, and SPSS (version-25) and R (version-3.5.1) were used for statistical analysis.

Table 1. Dataset details

Course	*Training set/enrolled (2016)	*Test set/enrolled (2017)
Introduction to programming	64/93	68/94
Algorithms and programming	170/248	172/258

*Students who participated in the problem-solving skills and course entry surveys, and completed formative assessment tasks and the final exam only selected for training and testing

Table 1 provides dataset details including course names and numbers of students enrolled in each year, as well as the numbers of students selected for training and test/unknown data sets. The data collected via LMS for the year 2016 ($n_1 = 64 + 170$) was used for feature selection, training and testing the models, for evaluation. The dataset collected in the year 2017 ($n_2 = 68 + 172$) was then employed for final testing (generalization performance) in order to propose models developed for the purpose of early warning systems.

3.2 Instruments and Assessments

Our study included two surveys, for self-assessment of problem-solving skills and prior programming knowledge, denoted the problem-solving skills and prior programming knowledge instruments, respectively. These surveys were conducted via the LMS at the start of the semester.

The problem-solving inventory (PSI) questionnaire contained 32 Likert-type questions to measure individual self-perception of problem-solving ability. The total score ranged from 32 to 192, with higher scores indicating poorer self-reported problem-solving skills. The PSI was developed by Heppner and Peterson [52]. The actual PSI questions were in English, devised by Heppner [52] translated into Finnish for *Algorithms and Programming* students, whose native language was Finnish. Moreover, this instrument was used in our prior study [53] to identify the relationship between PSI and academic performance of novice learners in an introductory programming course.

The prior programming knowledge (PPK) survey instrument was devised as part of a study by Veerasamy et al. [54] and comprised a 5-point (0 to 5) Likert scale of closed response questions for students. Each point was presented to students with a clear description to accurately self-assess their prior knowledge. In addition, students were asked to mention the names of programming languages they had learned and in which they had previously written at least 200 lines of code. Student responses to the PPK were crosschecked to measure the validity and reliability of the PPK survey. Later, these five points were collapsed in to three groups, identified as “0: no knowledge”, “1–2: basic knowledge” and “3–5: good knowledge”. Moreover, our prior study [54] confirmed the feasibility of using the PPK instrument to predict student academic performance.

A set of weekly homework exercises (HE) was provided for both courses, weekly, for 8 weeks. Each set of homework exercises averaged 5–10 questions, comprising objective type, code tracing, visualization and coding exercises. All exercises were delivered to students via the LMS, wherein they could submit their answers online, which were mostly automatically graded by LMS. The possible total HE scores for *Introduction to Programming* and *Algorithms and Programming* was 890 and 317, respectively.

Demo exercises (DE) for *Introduction to Programming* were dispatched to students weekly via the LMS, for 10 weeks throughout the semester. Each set of exercises had 4–7 coding questions. In a DE session (in the classroom), student solutions to questions for students who had completed them, and who were ready to submit, were discussed by the lecturer; selected students, subsequently selected randomly via the LMS, demonstrated their answers in class. No marks were awarded for class demonstrations. However, students who completed the DEs were instructed to enter their responses directly into the lecturer’s computer, to record the number of DEs completed by them [31]. The possible total DE score for *Introduction to Programming* was 750.

Tutorial exercises (TT) for *Algorithms and Programming* were provided to students weekly for 8 weeks throughout the semester. In a tutorial session students were given coding exercises via LMS to work online in the classroom. Students were allowed to submit their answers online, individually or as group submissions, which were automatically graded by ViLLE. However, a few coding exercises were manually graded by the lecturer, with scores entered into the LMS in order to assess students’ programming ability. The possible total TT score for *Algorithms and Programming* was 650. *Introduction to Programming* did not offer tutorials for students.

Both HE and DE were hurdles for *Introduction to Programming* with students having to attain at least 50% overall for HE and 40% over DE, in order to pass these components and the course. Similarly, all HE and TT were hurdles for *Algorithms and Programming*; students were required to secure at least 50% overall in each component and to have completed the end semester online-assignment in the LMS, to qualify to sit for the final exam. Both DE and TT sessions were conducted in the classroom, partially supervised and assisted by lecturer.

The final exam (FE) is an online summative assessment conducted at the end of the course of study. This final exam is conducted electronically via the LMS. The final exam is hurdle for *Introduction to Programming* and students are required to secure at least 50% (*) to pass this hurdle, in order to attain a grade for the course. However, the final exam is not a hurdle for *Algorithms and Programming*. Students attain 80% or more in the selected assessment components to get the maximum of two credit points and course grade 2. To obtain grades of 3 to 5, students must secure at least 50% or more in the selected assessment components and should get at least 62% or more in the final exam (Table 2).

The final exam grade (FEG) for the course was calculated based only on final exam scores. Table 2 shows the details of the grade calculation used for this study, to predict final exam grades for both courses.

Figure 1 and Fig. 2 present the grade-wise student distribution for *Introduction to Programming* and *Algorithms and Programming*. In this study, we defined students at-risk if they secured grades 0 or 1 in the final exam, and denoted their grade as “ZERO”.

Table 2. Grading criteria table: *Introduction to Programming & Algorithms and Programming*

Introduction to Programming		Algorithms and Programming	
Final exam marks	Grade	Final exam marks	Grade
0 to 49	0*	0 to 44	0*
50 to 59	1*	45 to 55	1*
60 to 69	2**	56 to 66	2**
70 to 79	3**	67 to 77	3**
80 to 92	4**	78 to 88	4**
93+	5**	89+	5**

*The actual grades 0 and 1 are considered as “at-risk” for this study and denoted as grade “ZERO”; **grades 2 to 5 are considered as “not-at-risk”

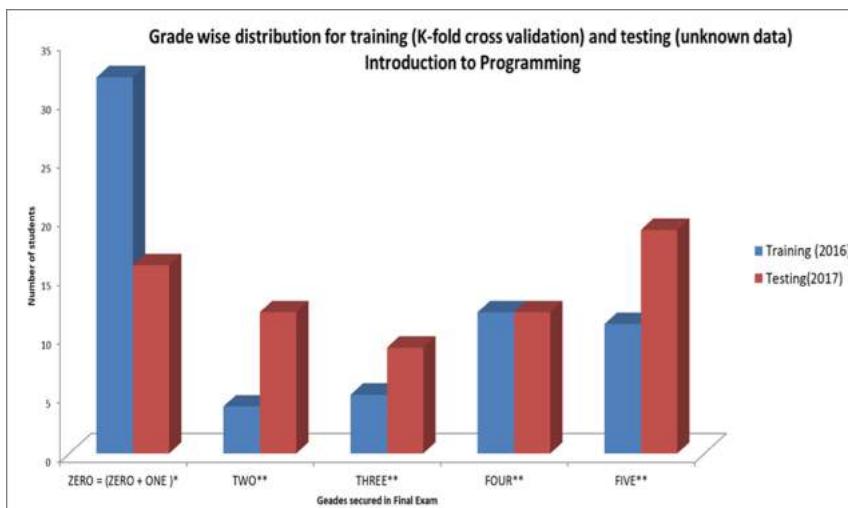
**Fig. 1.** Grade wise distribution chart- *Introduction to Programming*



Fig. 2. Grade wise distribution chart- *Algorithms and Programming*

This is because; students who secured a passing grade were not likely to succeed in subsequent courses. Hence, this study tags students who received a fail grade or a marginal pass as at-risk students, in order to check the at-risk student prediction accuracy of the model (Fig. 1 and 2).

Thirty-two (25 + 7) students secured grade “ZERO” in the year 2016 and 16 (11 + 5) in the year 2017 for *Introduction to Programming* (Fig. 1). Forty-four (30 + 14) students secured grade “ZERO” in the year 2016 and 32 (19 + 13) in the year 2017 for *Algorithms and Programming* (Fig. 2). Similarly, we defined students as not-at-risk who secured grades 2–5. In total, 32 students secured grades 2–5 in the year 2016 and 52 students in the year 2017, for *Introduction to Programming*; 126 students secured grades 2–5 in the year 2016 and 140 students in the year 2017, for *Algorithms and Programming*. The distribution of data is not unimodal.

3.3 Predictive Model

The goal of this study was to build a predictive model that most accurately predicts the desired output value for new input (course) data. Two (courses) x 15 predictive models were developed to measure the influence of selected predictor variables in prediction accuracy. In order to evaluate the prediction accuracy of the models, we used 5-fold cross-validation to ensure that the training and testing sets (year 2016) contained sufficient variation to arrive at unbiased results. In turn, this would avoid overfitting and allow us to establish how well the model generalized to unknown data (year 2017). We used the wrapper method (forward selection) to determine whether adding a specific feature would statistically improve the predictive performance of the model. In addition, the process was continued until all available variables were successively added to a model,

to identify the best set of variables for model development. The prediction accuracy of each of the 30 predictive models was examined by calculating the overall model prediction accuracy, the at-risk student prediction accuracy sensitivity and specificity, and area under the curve score (ROC curve), for each model. The following prediction accuracy measures were applied via R coding, to evaluate the performance of all models (in training and testing) to answer our research questions.

Model prediction accuracy (MPA): MPA was calculated as the number of correct predictions made by NBC, divided by the total number of actual values (and multiplied by 100) to get the prediction accuracy.

At-risk student prediction accuracy sensitivity (ATSE): The ATSE represents the percentage of at-risk students who are correctly identified by the model. The ATSE was calculated as:

$$ATSE = \frac{TAR}{TAR + FNR} \times 100$$

where TAR (True At-risk) is the number of predictions for grade “ZERO” that were correctly identified; and FNR (False Not At-risk) is the number of at-risk students who are incorrectly identified as not-at-risk students by the model. The ATSE value represents the percentage of at-risk students who are correctly identified by the model.

At-risk prediction accuracy specificity (ATSP): ATSP represents the percentage of not-at-risk students who are correctly identified by the model. The not-at-risk prediction accuracy was calculated as:

$$ATSP = \frac{TNR}{TNR + FAR} \times 100$$

where TNR” (True Not-At-risk) is the number of correctly identified predictions for grades “2–5; and FAR (False At-risk) is the number of not-at-risk students who are incorrectly identified as at-risk students by the model for all trials. The ATSP value represents the percentage of not-at-risk students who are correctly identified by the model.

Area under the ROC (receiver operating characteristics) curve (AUC): The AUC curve is a performance measurement for binary or multiclass classifiers. The AUC value lies between 0.5 and 1, inclusive, where 0.5 denotes a bad classifier and 1 denotes an excellent classifier. The higher the AUC, the better the model is at distinguishing between student at-risk and not-at-risk. The AUC was used to evaluate the diagnostic ability of the NBC model.

We defined the prediction accuracy of identifying at-risk and not-at-risk values, as follows: below 50% is considered poor; 50% - 69% moderate; 70%-79% good; and 80 and above as very good. As such, models with lowest predictive performances were dropped for tests on unknown data. The models returned the highest prediction accuracies (top three models X 2 courses) were then employed for testing unknown data (year 2017) for generalization.

4 Data Analysis and Results

We used SPSS for data pre-processing and RStudio to perform the NBC analysis (IBM, 2013). The data pre-processing was conducted as follows. First, all numerical data was scaled for standardization. For example, we converted the actual homework, demo and tutorial exercise scores (for the first four weeks of semester) to percentage scores. Next, the scaled data was stored as csv files to implement our NBC-based algorithms on these pre-processed datasets.

Table 3. The Models Developed for Feature Selection: Naive Bayes Classification - *Introduction to Programming*

Model#	Feature	Type	Model equation
#1	PSI	Cognitive variables	$\text{PSI} \rightarrow \text{FEG}$
#2	PPK		$\text{PPK} \rightarrow \text{FEG}$
#3	PSI, PPK		$\text{PSI, PPK} \rightarrow \text{FEG}$
#4	HE	Formative assessment tasks	$\text{HE} \rightarrow \text{FEG}$
#5	DE		$\text{DE} \rightarrow \text{FEG}$
#6	HE, DE		$\text{HE, DE} \rightarrow \text{FEG}$
#7	PSI, HE	Cognitive variables and formative assessment tasks	$\text{PSI, HE} \rightarrow \text{FEG}$
#8	PSI, DE		$\text{PSI, DE} \rightarrow \text{FEG}$
#9	PSI, HE, DE		$\text{PSI, HE, DE} \rightarrow \text{FEG}$
#10	PSI, PPK, HE		$\text{PSI, PPK, HE} \rightarrow \text{FEG}$
#11	PSI, PPK, DE		$\text{PSI, PPK, DE} \rightarrow \text{FEG}$
#12	PPK, HE		$\text{PPK, HE} \rightarrow \text{FEG}$
#13	PPK, DE		$\text{PPK, DE} \rightarrow \text{FEG}$
#14	PPK, HE, DE		$\text{PPK, HE, DE} \rightarrow \text{FEG}$
#15	PSI, PPK, HE, DE		$\text{PSI, PPK, HE, DE} \rightarrow \text{FEG}$

PSI: Problem solving skills; PPK: Prior programming knowledge; HE: Homework exercise; DE: Demo exercise; FE: Final exam; FEG: Final exam grade

We developed 15 predictive models for each of the two courses, with the following combinations of predictor variables to measure the differences between predictive capabilities of formative assessments and other variables to answer RQ1.

Models #1 – #15 were developed to predict final exam grades for *Introduction to Programming* and Models #16 – #30 were developed to predict final exam grades for *Algorithms and Programming*. Models #1 – #3 and #16 – #18 were developed using cognitive factors as input variables to predict final exam grades both courses. Models #4 – #6 and #19 – #21 were developed using formative assessment tasks as input variables to predict final exam grades for both courses. Models #7 – #15 and #22 – #30 were developed using both formative assessment tasks and cognitive factors as predictor variables to

predict student final exam grades for both courses. The models (#1 – #2, #4 – #5) and (#16 – #17, and #19 – #20) were developed with single features for both courses to examine the models' overall prediction accuracies, at-risk prediction accuracies, not-at-risk prediction accuracies and AUC results of those models, in order to determine the most valuable combination of predictors for model development. Tables 3 and 4 show NBC models developed for feature selection.

Table 4. The models developed for feature selection: Naive Bayes classification - *Algorithms and Programming*

Model#	Feature	Type	Model equation
#16	PSI	Cognitive variables	$\text{PSI} \rightarrow \text{FEG}$
#17	PPK		$\text{PPK} \rightarrow \text{FEG}$
#18	PSI, PPK		$\text{PSI}, \text{PPK} \rightarrow \text{FEG}$
#19	HE	Formative assessment tasks	$\text{HE} \rightarrow \text{FEG}$
#20	TT		$\text{TT} \rightarrow \text{FEG}$
#21	HE, TT		$\text{HE}, \text{TT} \rightarrow \text{FEG}$
#22	PSI, HE	Cognitive variables and formative assessment tasks	$\text{PSI}, \text{HE} \rightarrow \text{FEG}$
#23	PSI, TT		$\text{PSI}, \text{TT} \rightarrow \text{FEG}$
#24	PSI, HE, TT		$\text{PSI}, \text{HE}, \text{TT} \rightarrow \text{FEG}$
#25	PSI, PPK, HE		$\text{PSI}, \text{PPK}, \text{HE} \rightarrow \text{FEG}$
#26	PSI, PPK, TT		$\text{PSI}, \text{PPK}, \text{TT} \rightarrow \text{FEG}$
#27	PPK, HE		$\text{PPK}, \text{HE} \rightarrow \text{FEG}$
#28	PPK, TT		$\text{PPK}, \text{TT} \rightarrow \text{FEG}$
#29	PPK, HE, TT		$\text{PPK}, \text{HE}, \text{TT} \rightarrow \text{FEG}$
#30	PSI, PPK, HE, TT		$\text{PSI}, \text{PPK}, \text{HE}, \text{TT} \rightarrow \text{FEG}$

PSI: Problem solving skills; PPK: Prior programming knowledge; HE: Homework exercises; TT: Tutorial exercise; FE: Final exam; FEG: Final exam grade

We were interested in identifying the most valuable features in predicting student performance for both courses. As such, we examined the resultant AUC for each feature when used for model development and final testing. Figure 3 and Fig. 4 present the resultant AUC for training and unknown (testing) datasets, when using each feature individually to predict student final exam grades for both courses.

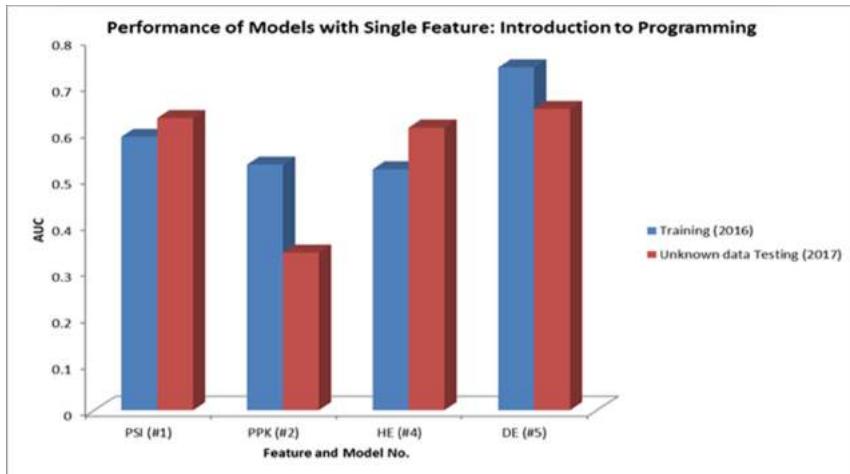


Fig. 3. Performance of models with single feature: *Introduction to Programming*

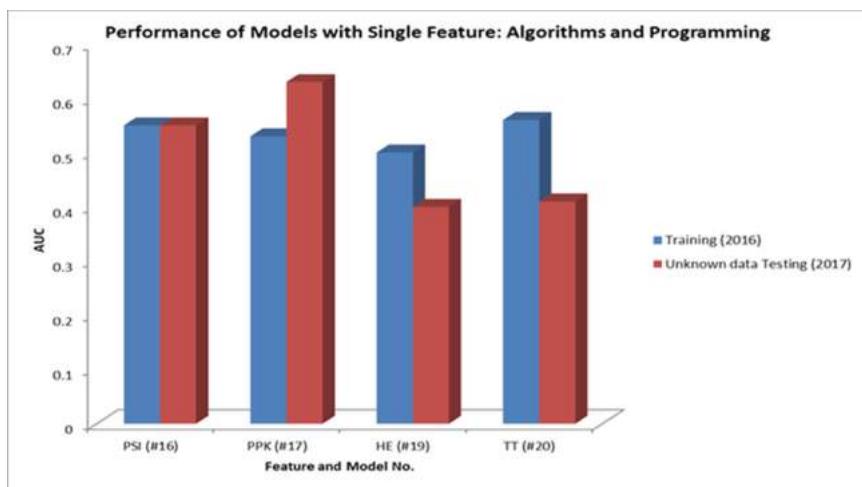


Fig. 4. Performance of models with single feature: *Algorithms and Programming*

The DE feature returns the best AUC out of all other features, and followed by PSI. This implies that DE and PSI are powerful predictors of student performance in *Introduction to Programming* (Fig. 3). On the other hand, the cognitive features (PPK and PSI) returned the best AUC out of all other features for *Algorithms and Programming*, although AUC results on the unknown dataset were varied (Fig. 4).

As noted, one of the objectives of this study was to determine whether our model(s) could be used as early warning system(s) for instructors to identify students in need of academic support (RQ2). Therefore, we selected top three models (for both courses) that returned higher prediction accuracies in the year 2016 (Table 5). Then these selected

models were employed on the dataset collected in the year 2017 (unknown data) to determine how well these models would work on unknown data (Table 6) and to propose the models that with good predictive power as early warning systems.

Table 5. The overall and at-risk student prediction accuracies on training set: 2016

Model#	Overall prediction accuracy (MPA)	At-risk prediction accuracy (ATSE)	Not-At-Risk prediction accuracy (ATSP)	Area under the curve (AUC)	Course
#8	79.69	81.25	78.12	0.79	<i>Introduction to Programming</i>
#11	75.00	81.25	68.75	0.78	
#5	76.56	78.12	75.00	0.74	
#29	65.21	61.36	69.05	0.65	<i>Algorithms and Programming</i>
#30	65.26	59.09	71.43	0.64	
#28	61.24	61.36	61.11	0.59	

Table 5 shows the top three selected models for each of the two courses for the year 2016, which had higher prediction accuracies (AUC scores). The K-fold cross-validation results for the training set revealed that model #8 with PSI and DE only, as predictors, returns the best prediction accuracy (MPA: 80%, ATSE:81, ATSP:78, AUC:0.79) for *Introduction to Programming*. Similarly, model #29 developed with continuous assessments tasks (HE and TT) and the cognitive variable (PPK) only, as predictors, returned the best prediction accuracy for *Algorithms and Programming* (MPA: 65%, ATSE: 61%, ATSP: 69%, AUC: 0.65).

Table 6. Top three models' performance on test dataset for the year 2017

Model#	Overall prediction accuracy (MPA)	At-risk prediction accuracy (ATSE)	Not-At-Risk prediction accuracy (ATSP)	Area under the curve (AUC)	Course
#8	70.91	93.75	48.08	0.66	<i>Introduction to Programming</i>
#11	67.79	87.50	48.08	0.63	
#5	68.99	93.75	44.23	0.65	
#29	49.65	0	99.29	0.41	<i>Algorithms and Programming</i>
#30	49.65	0	99.29	0.41	
#28	49.65	0	99.29	0.41	

Table 6 shows the models' unknown data test results (year 2017) of top three models for each of the two courses. The results were mixed. On average, the at-risk prediction

accuracy for identifying students who needed support for *Introduction to Programming*, was 91.67% (Models #8, #11, and #5) in compliance with AUC scores (0.66). However, the not-at-risk prediction accuracy, for identifying student who did not need support, was statistically poor (46.80%). On the other hand, on average, the at-risk prediction accuracy, for identifying students who needed support for *Algorithms and Programming* was 0% (Models #29, #30, and #28), which is statistically insignificant and addressed below.

5 Discussion

The main objective of this study was to construct a predictive model with as few predictor variables to predict final programming exam performance of students in order to identify at-risk students early. The validation and unknown data test results for models (Fig. 3) with a single feature as predictors for *Introduction to Programming* revealed that demo exercises were the most influential factor (AUC: 0.65) in determining student final exam performance. On the other hand, prior programming knowledge returned the best AUC (0.63) and may serve as a predictor of student success in *Algorithms and Programming* (Fig. 4). These results imply that models developed and tested with combination of demo and problem solving skills may yield better prediction accuracies than models developed with other features for *Introduction to Programming*. Similarly, the unknown dataset results for models with single features as predictors for *Algorithms and Programming* (Fig. 4) revealed that models with cognitive features (prior programming knowledge and problem solving skills) may yield better prediction accuracies than other models.

Our statistical results of prediction accuracies computed across all K-fold cross-validation (training) and unknown data tests for *Introduction to Programming* yielded good results (Tables 5 and 6). That is, it is possible to identify at-risk students in the first four weeks, based on student prior programming knowledge, problem solving skills, and formative assessment results. Hence, these results answered our RQ1. The overall model prediction accuracy and at-risk prediction accuracy of the top three models (year 2016) selected for *Introduction to Programming* were very good and congruent with single feature based model results (Fig. 3 and Table 5). In addition, the unknown data test results (year 2017) reveal that the models selected based on high prediction accuracies can be proposed as early warning systems for *Introduction to Programming* (Table 6).

On the other hand, the unknown dataset test results on identifying at-risk students for *Algorithms and Programming* produced insignificant results (Table 6) and were not congruent with results of models tested with single features (Fig. 4). These results raised the following points. First, unlike for *Introduction to Programming*, the grade computation for *Algorithms and Programming* is slightly different. The final exam need not be sat to pass *Algorithms and Programming*, which would have influenced the predictive performance of the models (#28–#30). Second, registration to attend the final exam is allowed as late as the last lecture week, which would have affected the student performance in the final exam. Moreover, if a student did not secure 50% or more in the final exam they were still able to attain at least a grade 1 score, by securing 80%–94% in the formative assessment tasks. Third, as per standard rules, students could opt to sit the final exam to improve their final course grades despite the scores* they attained

in selected formative assessments. However, post preliminary anecdotal results showed that students who secured adequate scores in formative assessments (*50% or more but below 95%), who were therefore eligible to sit for final exam, achieved higher grades than students who achieved grade 1 or 2 via formative assessment scores. Therefore, the differences between student final exam grades and grades achieved through selected formative assessment scores warrants further analysis, in order to improve the model's predictive performance for *Algorithms and Programming*.

Moreover, these results persuaded us to redefine our research question, RQ2: (i) Might our proposed model with these predictor variables be deployed in an early warning system to support instructors and students? (ii) Might our proposed model be transformed as a generic predictive model for other courses that with continuous formative assessments and final exam, to predict student performance early in the semester? As noted, several studies attempted to establish early warning systems to identify at-risk students and allow for more timely pedagogical interventions [46–48]. In the same vein, we also attempted to develop a model with the aim of using it for all introductory programming courses. The at-risk prediction accuracy for at-risk student training and unknown data test results (81% and 94%) (#8) of this study supports the contention that our proposed model may be deployed as an early warning system to predict students who need early assistance. In addition, these (Tables 5 and 6) results also revealed that, it is possible to predict student who need support early based on problem-solving skills and formative assessment (demo exercise) scores, secured in the first four weeks of the semester (Fig. 3 and Table 6). Hence, these results imply that our model may be adapted as an early warning system in programming courses assessed with continuous assessment and final exam components, to predict student academic performance and to identify students who need support.

The model may be used by instructors to categorize students as, for example, “at-risk”, “marginal”, “average”, “good”, “very good”, and “excellent”, based on predicted final exam grades, and accordingly to reshape their pedagogical practices. In addition, instructors may deploy this model as part of their student academic monitoring system to get actionable data for analysis and to support students in personalised ways. For example, after identifying less-motivated learners via this model, instructors may counsel them about strategies to improve their performance.

Students too may use the model(s) in the following ways. First, the results of these models may be delivered as real-time feedback to learners, to persuade or encourage them to devote attention to specific learning activities, in order to improve their performance before they reach a critical juncture. Second, students may perceive these results as ongoing performance indicators to shore up their own learning goals to improve learning and academic performance. Third, the detection of early warning signs could persuade students towards alleviating their learning. Therefore, we conclude that the results of these models may help students to alter their learning behaviors, and to better understand their performances, shortcomings and successes.

As noted earlier, most previous studies to develop predictive models reported that their models have generalizability problems, due to the following factors: (a) unsuitability of research methods due to technical, cultural, or ethical reasons; (b) course specific predictor variables for the model developed; and (c) cognitive factors. Keeping these

problems in mind, we developed our model with transformable predictor variables to test for other courses and on unknown data. Take first the student perceived general problem-solving skills. Many studies have reported the importance of measuring student general problem-solving abilities [16, 55]. Our research findings (Fig. 3) also suggested that being aware of the problem-solving skills of incoming freshmen would help instructors to review their problem-solving based instructional methods, to develop and to enhance students' metacognitive, computational thinking skills [53]. Based on these results, we recommend that student perceived problem-solving skills should most likely be included as one of the predictive variables in mathematical models for predicting student performance.

Consider now the second variable, prior programming knowledge. Although student perceived problem-solving ability plays an important role in student learning, prior subject knowledge remains the central variable in student learning and performance [56]. In addition, the results of this study and our past work [54] suggest that prior knowledge can be considered as appropriate for predicting at-risk students (Tables 5 and 6). However, our unknown data test results of models (#28-#30) produced insignificant results, which needs further analysis. Despite these mixed results this variable may be adapted to predict student academic performance.

As noted, predictive models developed for predicting student performance have at least one assessment task as a predictor variable. The overall prediction accuracy of the models (Tables 5 and 6) confirms that a formative assessment task can have predicting influence of student performance, early in the semester. Therefore, we conclude that formative assessment tasks may be considered as a significant predictor for measuring student progress early. However, the selection of predictor variables associated with assessment tasks varies from course to course. For example, in this study we chose homework and demo/tutorial exercises for our predictive model, based on earlier research findings [31] to predict student performance. However, our feature selection and unknown data test results suggest that the models developed with in-classroom assessments (DE/TT) have higher prediction accuracies (MPA, ATSE and ATSP) than models developed with outside-classroom assessments, a result that needs further analysis. Therefore, it is important to select suitable or discipline specific formative assessment tasks to measure student-learning outcomes, to track ongoing performance of students. However, it should be noted that courses that do not have continuous assessments and only a final exam may use cognitive factors only, as predictor variables as post analysis.

Based on the past research findings and results of our study (Fig. 3 and Tables 5 and 6 for *Introduction to Programming*) we have argued that our proposed generic predictive model may be deployed for other programming and non-programming courses, if the goal of instructor is to predict student performance and to identify low motivated learners early in the semester.

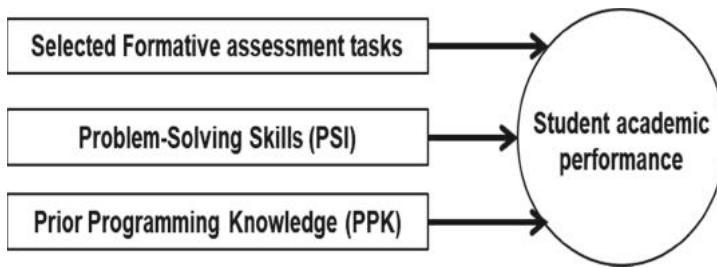


Fig. 5. Generic predictive model for student performance prediction

6 Conclusions, Limitations and Future Work of the Study

Our research study showed that student perceived problem-solving skills, prior programming knowledge, and formative assessment tasks were significant in predicting student final exam grades for *Introduction to Programming*. However, the unknown dataset test results for *Algorithms and Programming* were statistically insignificant. The results showed that student perceived problem-solving skills, prior programming knowledge and formative assessments, captured in a predictive model, was a good fit of the data. Furthermore, the models developed with a combination of in-class assessment variables and cognitive features yielded better at-risk prediction accuracies than other models, developed with a combination of take-home assessment variables. The overall success of the models was good, which persuaded us to update our research questions (Tables 5 and 6). The model may be used as an early warning system for instructors to identify students needing early assistance. Therefore, we presented the generic form of predictive model (Fig. 5), and how this model could be applied and developed in future research, for early identification of at-risk students in other courses that use continuous assessment tasks and a final exam. The results of the study will be used in future work, to build an accurate predictive model for student academic performance in other programming courses.

Despite our promising results, this study has several limitations that influence the overall generalizability and interpretation of the findings. First, the data used in this study was collected within one institution. From a statistical point of view, it is true that sample size influences research outcomes. However, a number of studies on predictive models suggest that sample size for analysis is often determined by the research question or the analysis intended, although bigger samples are better to define the reliability and validity of the research [57]. As such, in future we plan to include data from multiple courses collected over multiple semesters. Second, although this study used multiclass classification, we mainly focused on examining at-risk class accuracies and did not analyse other classes of the dataset prediction accuracies individually. However, as noted (Fig. 1 and 2), all other classes (grades 2 to 5) were considered as not-at-risk for this study to proceed further. Third, we measured student prior programming knowledge on a broad level. Therefore, it is unknown whether the participants responded honestly to the survey. Fourth, although the average prediction accuracy on test data was good, the predictive performance of the model might be influenced by the course assessment nature, which means that the resulting model may not work well to predict unknown

data. Fifth, we used the first four weeks of assessment for analysis. However, learning is dynamic and so a learner might do well in the beginning weeks of the semester and may not perform well in second half of the semester. Hence, there is a need to monitor and track student progress throughout the course period, in order to be better informed about providing continuous academic support. Therefore, we plan to extend our study to predict student performance based on assessment results collected in different weeks of the course, to determine if these results better inform instructors to monitor and evaluate student-learning outcomes. For example, if the duration of the course period is 12 weeks, then the prediction will be deployed in three cycles based on the first, middle, and final four weeks of the course.

In spite of these limitations, we contend that our models can be applied to other courses with continuous formative assessments and a final exam to predict student performance. Moreover, in this work, we proposed a prediction methodology using Naive Bayes classification for early identification of students that need support. This generic model introduced in this research paper provides a guide for future work. For example, to identify at-risk students in accounting, the predictor variables used in this study may be adapted as generic predictors to fit course specific data mining predictive models, for student performance. That is, prior programming knowledge can be transformed as prior knowledge in accounting to measure student prior accounting knowledge, with selected formative assessment tasks to predict student performance. The possible research, notionally, is predicting accounting student performance using prior accounting knowledge and selected ongoing formative assessment task results. Similarly, problem-solving skills may be used to measure student perceived problem-solving abilities of accounting students. Moreover, this model may be deployed to seek answers to the following research questions: (i) Does prior knowledge influence student learning? (ii) How does student prior knowledge in the topic/subject impact classroom teaching and learning? (iii) What kind of support activities might be provided to at-risk students identified by this model? (iv) Does the prediction accuracy of the model vary significantly based on the predictive algorithms used? Apart from the above, the proposed model of this study might be used for further research including some more significant factors that are more widely used in predictive models for predicting student performance. For example, psychological variables such as self-belief, self-regulated learning and self-efficacy, as well as and educational variables, such as prior GPA, and other academic variables available obtainable from the LMS may be included to enhance the prediction accuracy of model.

Acknowledgments. The authors wish to thank all members of ViLLE research team group and Department of Future technologies, University of Turku, for their comments and support that greatly improved the manuscript. This research was supported fully by a University of Turku, Turku, Finland.

References

1. Ali, A., Smith, D.: Teaching an introductory programming language. *J. Inf. Technol. Educ.: Innov. Pract.* **13**, 57–67 (2014)

2. Holvikivi, J.: Conditions for successful learning of programming skills. In: Reynolds, N., Turcsányi-Szabó, M. (eds.) KCKS 2010. IAICT, vol. 324, pp. 155–164. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15378-5_15
3. Watson, C., Li, F.W.B. Failure Rates in introductory programming revisited. In Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education (Uppsala 2014), pp. 39–44. Association of Computing Machinery (2014)
4. Bennedsen, J., Caspersen, M.: Failure rates in introductory programming: 12 years later. ACM Inroads **10**(2), 30–36 (2019)
5. Castro-Wunsch, K., Ahadi, A., Petersen, A.: Evaluating neural networks as a method for identifying students in need of assistance. In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (Seattle 2017), pp. 111–116. ACM (2017)
6. Conijn, R., Snijders, C., Kleingeld, A., Matzat, U.: Predicting student performance from LMS data: a comparison of 17 blended courses using moodle LMS. IEEE Trans. Learn. Technol. **10**(1), 17–29 (2017)
7. Liao, S.N., Zingaro, D., Alvarado, C., Griswold, W.G., Porter, L.: Exploring the value of different data sources for predicting student performance in multiple CS courses. In: Proceedings of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, MN, USA, pp. 112–118. ACM (2019)
8. Pawlowska, D.K., Westerman, J.W., Bergman, S.M., Huelsman, T.J.: Student personality, classroom environment, and student outcomes: a person–environment fit analysis. Learn. Individ. Differ. **36**, 180–193 (2014)
9. Costa, E.B., Fonseca, B., Santana, M.A., Araújo, F.F., Rego, J.: Evaluating the effectiveness of educational data mining techniques for early prediction of students’ academic failure in introductory programming courses. Comput. Hum. Behav. **73**, 247–256 (2017)
10. Roberts, S.A.: Parsimonious modelling and forecasting of seasonal time series. Eur. J. Oper. Res. **16**, 365–377 (1984)
11. Vandekerckhove, J., Matzke, D., Wagenmakers, E.-J.: Model comparison and the principle of parsimony. In: Busemeyer, J.R., et al. (eds.) The Oxford Handbook of Computational and Mathematical Psychology. Oxford University Press, New York (2015)
12. D’zurilla, T.J., Nezu, A.M., Maydeu-Olivares, A.: Social problem solving: theory and assessment. In: Chang, E.C., et al. (eds.) Social problem Solving: Theory, Research, and Training. American Psychological Association, Washington, DC (2004)
13. White, H.B., Benore, M.A., Sumter, T.F., Caldwell, B.D., Bell, E.: What skills should students of undergraduate biochemistry and molecular biology programs have upon graduation? Biochem. Mol. Biol. Educ. **41**(5), 297–301 (2013)
14. Kappelman, L., Jones, M.C., Johnson, V., McLean, E.R., Boonme, K.: Skills for success at different stages of an IT professional’s career. Commun. ACM **59**(8), 64–70 (2016)
15. Heppner, P.P., Krauskopf, C.J.: An information-processing approach to personal problem solving. Counsell. Psychol. **15**(3), 371–447 (1987)
16. Adachi, P.J.C., Willoughby, T.: More than just fun and games: the longitudinal relationships between strategic video games, self-reported problem solving skills, and academic grades. J. Youth Adolesc. **42**(7), 1041–1052 (2013)
17. Bester, L.: Investigating the problem-solving proficiency of second-year quantitative techniques students: the case of Walter Sisulu University. University of South Africa, Pretoria (2014)
18. Marion, B., Impagliazzo, J., St. Clair, C., Soroka, B., Whitfield, D.: Assessing computer science programs: what have we learned. In: SIGCSE 2007 Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education, Covington, Kentucky, USA, pp. 131–132. ACM (2007)
19. Ring, B.A., Giordan, J., Ransbottom, J.S.: Problem solving through programming: motivating the non-programmer. J. Comput. Sci. Coll. **23**(3), 61–67 (2008)

20. Uysal, M.P.: Improving first computer programming experiences: the case of adapting a web-supported and well-structured problem-solving method to a traditional course. *Contemp. Educ. Technol.* **5**(3), 198–217 (2014)
21. Svinicki, M.: What they don't know can hurt them: the role of prior knowledge in learning. POD network, Nederland, Colorado (1993)
22. Hailikari, T.: Assessing university students' prior knowledge implications for theory and practice. Helsinki (2009)
23. Watson, C., Li, F.W.B., Godwin, J.L.: No tests required: comparing traditional and dynamic predictors of programming success. In: Proceedings of the 45th ACM Technical Symposium on Computer Science Education, pp. 469–474. ACM (2014)
24. Longi, K.: Exploring factors that affect performance on introductory programming courses. University of Helsinki, Helsinki (2016)
25. Hsu, W.C., Plunkett, S.W.: Attendance and grades in learning programming classes. In: Proceedings of the Australasian Computer Science Week Multi Conference, Canberra (2016)
26. Sabin, M., Alrumaih, H., Impagliazzo, J., Lunt, B., Zhang, M.: Information technology curricula 2017: curriculum guidelines for baccalaureate degree programs in information technology. ACM and IEEE, New York (2017)
27. Bloom Benjamin, S., Hastings, J.T., Madaus, G.F.: *Handbook on Formative and Summative Evaluation of Student Learning*. McGraw-Hill Book Company, New York (1971)
28. Lau, A.M.S.: 'Formative good, summative bad?' – a review of the dichotomy in assessment literature. *J. Furth. High. Educ.* **40**(16), 509–525 (2016)
29. VanDeGrift, T.: Supporting creativity and user interaction in CS 1 homework assignments. In: 46th ACM Technical Symposium on Computer Science Education, Kansas City, pp. 54–59. ACM (2015)
30. Rajoo, M., Veloo, A.: The relationship between mathematics homework engagement and mathematics achievement. *Aust. J. Basic Appl. Sci.* **9**(28), 136–144 (2015)
31. Veerasamy, A.K., D'Souza, D., Lindén, R., Kaila, E., Laakso, M.-J., Salakoski, T.: The impact of lecture attendance on exams for novice programming students. *Int. J. Mod. Educ. Comput. Sci. (IJMECS)* **8**(5), 1–11 (2016)
32. Fan, H., Xu, J., Cai, Z., He, J., Fan, X.: Homework and students' achievement in math and science: A 30-year meta-analysis, 1986–2015. *Educ. Res. Rev.* **20**, 35–54 (2017)
33. Fujinuma, R., Wendling, L.: Repeating knowledge application practice to improve student performance in a large, introductory science course. *Int. J. Sci. Educ.* **37**(17), 2906–2922 (2015)
34. Thong, L.W., Ng, P.K., Ong, P.T., Sun, C.C.: Performance analysis of students learning through computer-assisted tutorials and item analysis feedback learning (CATIAF) in foundation mathematics. *Herald NAMSCA*, vol. 1, p. 1 (2018)
35. Ahadi, A., Lister, R., Haapala, H., Vihavainen, A.: Exploring machine learning methods to automatically identify students in need of assistance. In: Proceedings of the Eleventh Annual International Conference on International Computing Education Research, Omaha, Nebraska, USA, pp. 121–130. ACM (2015)
36. Porter, L., Zingaro, D., Lister, R.: Predicting student success using fine grain clicker data. In: Proceedings of the Tenth Annual Conference on International Computing Education Research, Glasgow, Scotland, United Kingdom, pp. 51–58. ACM (2014)
37. Quille, K., Bergin, S.: Programming: predicting student success early in CS1. A re-validation and replication study. In: Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education, Larnaca, Cyprus, pp. 15–20. ACM (2018)
38. Liao, S., Zingaro, D., Thai, K., Alvarado, C., Griswold, W., Porter, L.: A robust machine learning technique to predict low-performing students. *ACM Trans. Comput. Educ.* **19**(3), 18:1–18:19 (2019)

39. Liao, S.N., Zingaro, D., Laurenzano, M.A., Griswold, W.G., Porter, L.: Lightweight, early identification of at-risk CS1 students. In: Proceedings of the 2016 ACM Conference on International Computing Education Research, Melbourne, VIC, Australia, pp. 123–131. ACM (2016)
40. Hamoud, A.K., Humadi, A.M., Awadh, W.A., Hashim, A.S.: Students' success prediction based on Bayes algorithms. *Int. J. Comput. Appl.* **178**(7), 6–12 (2017)
41. Devasia, T., Vinushree, T.P., Hegde, V.: Prediction of students performance using educational data mining. In: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), Ernakulam, pp. 91–95. IEEE (2016)
42. Agrawal, H., Mavani, H.: Student performance prediction using machine learning. *Int. J. Eng. Res. Technol. (IJERT)* **4**(3), 111–113 (2015)
43. Bergin, S., Mooney, A., Ghent, J., Quille, K.: Using machine learning techniques to predict introductory programming performance. *Int. J. Comput. Sci. Softw. Eng.* **4**(12), 323–328 (2015)
44. Borrà, S., Di Ciaccio, A.: Measuring the prediction error. A comparison of cross-validation, bootstrap and covariance penalty methods. *Comput. Stat. Data Anal.* **54**, 2976–2989 (2010)
45. Macfadyen, L.P., Dawson, S.: Mining LMS data to develop an “early warning system” for educators: a proof of concept. *Comput. Educ.* **54**, 588–599 (2009)
46. Krumm, A., Joseph Waddington, R., Teasley, S., Lonn, S.: A learning management system-based early warning system for academic advising in undergraduate engineering. In: Larusson, J.A., White, B. (eds.) *Learning Analytics*, pp. 103–119. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-3305-7_6
47. Arnold, K.E., Pistilli, M.D.: Course signals at Purdue: using learning analytics to increase student success. In: Proceedings of the 2nd International Conference on Learning Analytics and Knowledge, Vancouver, British Columbia, Canada, pp. 267–270. ACM (2012)
48. Pistilli, M., Willis, J., Campbell, J.: Analytics through an institutional lens: definition, theory, design, and impact. In: Larusson, J.A., White, B. (eds.) *Learning Analytics*, pp. 79–102. Springer, New York (2014). https://doi.org/10.1007/978-1-4614-3305-7_5
49. Ya-Han, H., Lo, C.-L., Shih, S.-P.: Developing early warning systems to predict students' online learning performance. *Comput. Hum. Behav.* **36**, 469–478 (2014)
50. Pedraza, D.A.: The relationship between course assignments and academic performance: an analysis of predictive characteristics of student performance. Texas Tech University (2018)
51. Marbouti, F., Diefes-Dux, H.A., Madhavan, K.: Models for early prediction of at-risk students in a course using standards-based grading. *Comput. Educ.* **103**, 1–15 (2016)
52. Heppner, P.P., Petersen, C.H.: The development and implications of a personal problem-solving inventory. *J. Counsell. Psychol.* **29**(1), 66–75 (1982)
53. Veerasamy, A.K., D'Souza, D., Lindén, R., Laakso, M.-J.: Relationship between perceived problem-solving skills and academic performance of novice learners in introductory programming courses. *J. Comput. Assist. Learn.* **35**(2), 246–255 (2019)
54. Veerasamy, A.K., D'Souza, D., Linden, R., Laakso, M.-J.: The impact of prior programming knowledge on lecture attendance and final exam. *J. Educ. Comput. Res.* **56**(2), 226–253 (2018)
55. Özyurt, Ö.: Examining the critical thinking dispositions and the problem solving skills of computer engineering students. *Eurasia J. Math.* **11**, 2 (2015)
56. Chakrabarty, S., Martin, F.: Role of prior experience on student performance in the introductory undergraduate CS course. In: SIGCSE 2018 Proceedings of the 49th ACM Technical Symposium on Computer Science Education, Baltimore, Maryland, USA, pp. 1075–1075. ACM (2018)
57. Austin, P., Steyerberg, E.: The number of subjects per variable required in linear regression analyses. *J. Clin. Epidemiol.* **68**(6), 627–636 (2015)



Youden's J and the Bi Error Method

MaryLena Bleile^(✉)

Statistical Science, Southern Methodist University, Dallas, USA
mbleile@smu.edu

Abstract. Incorrect usage of p -values, particularly within the context of significance testing using the arbitrary .05 threshold, has become a major problem in modern statistical practice. The prevalence of this problem can be traced back to the context-free 5-step method commonly taught to undergraduates: we teach it because it is what is done, and we do it because it is what we are taught. This holds particularly true for practitioners of statistics who are not formal statisticians. Thus, in order to improve scientific practice and overcome statistical dichotomania, an accessible replacement for the 5-step method is warranted. We propose a method foundational on the utilization of the Youden Index as a potential decision threshold, which has been shown in the literature to be effective in conjunction with neutral zones. Unlike the traditional 5-step method, our 5-step method (the Bi Error method) allows for neutral results, does not require p -values, and does not provide any default threshold values. Instead, our method explicitly requires contextual error analysis as well as quantification of statistical power. Furthermore, and in part due to its lack of usage of p -values, the method sports improved accessibility. This accessibility is supported by a generalized analytical derivation of the Youden Index.

Keywords: Youden Index · p-Value · Type II error

1 Introduction

The problematic nature of blindly executing hypothesis tests using the $p < .05$ has been clearly illustrated: the problem received official recognition by the ASA in 2016, culminating in the recent release of a special edition of *The American Statistician*, which featured 43 papers on the topic [6, 16]. Subsequently, the National Institute of Statistical Science hosted a webinar on alternatives to the p -value, which was followed up in November with a more in-depth discussion featuring three experts in the field: Jim Berger, Sander Greenland, and Robert Matthews [7, 8]. Cobb points out the cyclic nature of the problem: “We teach it because it’s what we do, we do it because it’s what we teach” [16]. A recent paper by Cassidy, et al. even went so far as to show that 89% of psychology textbooks that attempt to explain statistical significance, do so incorrectly [2]. Furthermore, in some cases, misinterpretation of statistical significance (or non-significance) can be fatal to those who would otherwise benefit from the results

of the research: earlier this year, a study which investigated the effectiveness of Remdesivir as a treatment for COVID-19 incorrectly interpreted their lack of statistically significant results to mean that “Remdesivir had no effect”, when in reality the study was merely underpowered; the number of subjects was not sufficient to reduce the background noise in the data enough for the signal to be perceivable [12]. A later study with greater statistical power showed evidence that Remdesivir is, in fact, an effective treatment for the coronavirus [1].

Yet in spite of the extensive literature on the adverse effects of procedural, context-blind statistical hypothesis testing with $p < .05$, we continue to teach (and sometimes practice) the 5-step Null-Hypothesis Significance Testing (NHST) method outlined in Algorithm 1.

Algorithm 1. NHST Method

- 1: Specify the null hypothesis
 - 2: Specify the alternative hypothesis
 - 3: Set the significance level (usually $\alpha = .05$)
 - 4: Calculate the test statistic and the corresponding p-value
 - 5: If the p-value is less than the significance level, then reject the null hypothesis and conclude statistical significance. Otherwise, fail to reject the null hypothesis and conclude statistical non-significance.
-

Of course, the field of statistical methodology is rich and extensive beyond this. Specifically, hypothesis testing with neutral zones has shown to be effective. The idea behind this strategy is to allow for a third option aside from rejecting or failing to reject the null hypothesis H_0 : *inconclusive results*. However, many of the methods from the statistical literature lack the simplicity and algorithmic structure of the existing 5-step method, which is perhaps what contributes to its attractiveness for teaching and for use by non-statisticians. To quote Wasserstein, Schirm, & Lazar, “Don’t is not enough”: it is not enough to simply ban scientists and statisticians from using or teaching the existing 5-step method, leaving a gap in the curriculum. In order to fill this gap, we propose a replacement 5-step method for use in place of the above [17].

Statistical hypothesis tests can be considered as binary classifiers between the null and alternative hypotheses. The Receiver Operating Characteristic curve, which plots sensitivity vs 1-specificity is of special interest when discussing classifiers. (Here, sensitivity is the propensity of the test to reject the null given that the null is actually false, whereas specificity is the propensity of the test to *fail* to reject the null, given that the null is actually true). Naturally, adding these two quantities together results in a value analogous to the chance of making a correct decision. Note, both sensitivity and specificity depend only on what decision threshold one uses to classify. The Youden Index is defined as $J = \max_c(Sensitivity(c) + Specificity(c) - 1)$, where c is the decision threshold [19]. That is, J is the decision threshold which maximizes the aggregated sensitivity and specificity of the classifier (see Fig. 1).

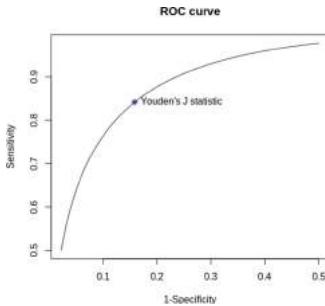


Fig. 1. An ROC curve illustrating the Youden Index.

In other words, J is the decision threshold which minimizes the additive Type I and Type II error (where Type I error, denoted α , is the probability of falsely rejecting the null, and Type II error, denoted β , is the probability of failing to reject when the null is false). That is $J = \min_{c} \arg_c(\alpha + \beta)$ (note, α, β are both functions of the decision threshold c). Youden's Index has been widely used in the context of hypothesis testing with neutral zones [9].

Algorithm 2 outlines the proposed method. We call this method the “Bi Error Method” for the two following reasons:

1. Youden's J minimizes two error terms
2. Subjective analysis of two error rates is required

Algorithm 2. The Bi Error Method

- 1: Identify the null and alternative sampling distributions
 - 2: Set reasonable upper thresholds for Type I and Type II error (α, β , respectively).
 - 3: Find (or approximate) the Youden Index (J)
 - 4: If the values of α, β , (and, implicitly, $\zeta(J)$) are reasonable, proceed to 5. Else, if any one of the above is too high, the results are inconclusive.
 - 5: If α, β are reasonable and the observed test statistic is more extreme than x_0 , then reject H_0 . Else, fail to reject H_0 .
-

We provide five of the most compelling reasons why the proposed method is attractive as a replacement, at least for pedagogical purposes.

Accessibility. Unlike some of the more sophisticated methodology used in the statistical literature, our method has the same 5-step, algorithmic structure as the existing method.

No p-Values. The pervasiveness and negative effects of misinterpretations of p-values in the academic literature is well-documented [2,4,5,14,16]. Our method eliminates the need for their use, which serves to potentially increase the accuracy of statistical practice in scientific research.

Contextual Error Analysis. One of the ASA’s recommendations for hypothesis testing was that context should be considered in the analysis [16]. Yet, we still run into issues where the hypothesis testing is blindly performed with no attention to context, which has led to potentially disastrous results [4, 12, 14]. Our method propagates “Thoughtful research” as discussed in the literature, by including an explicit step involving contextual analysis, with no default thresholds for any of the values analyzed [17].

Inconclusive Results Option. The method propagates the acceptance of uncertainty in another way. In congruence with methods that have been shown to be effective in the literature, it explicitly allows for a third option besides rejecting or failing to reject the null hypothesis: *inconclusivity* [9, 11].

Quantitative Incorporation of Type II Error. One of the major issues with the statistical hypothesis testing method currently in place, is that it does not take into account the probability of falsely failing to reject: aka Type II error [4, 12]. In contrast, through the utilization of the Youden index, this method explicitly incorporates this quantitatively into the analysis.

The remainder of the paper is structured as follows: in Sect. 2 we present the methods utilized in order to critically evaluate the performance Bi Error Method through simulation, as compared to the NHST method. Section 3 includes a brief description of the results of this evaluation, as well as the analytic results. In Sect. 4 we discuss potential criticisms of the method, as well as analytic comparison of the expected accuracy of the Bi Error Method as compared to the NHST method. Finally, we investigate applications in cases where the analytical result does not apply.

2 Methods

Since our method involves contextual, subjective error analysis, it is impossible to simulate its true performance. We can, however, get a general idea of what its lower bound on performance might be by simulating the hypothesis testing scenario and dichotomously classifying results based on whether they are larger than the Youden index.

It is notable that since we are utilizing the Type II error in the test procedure, hypothesis tests must be performed with a point alternative. This is desirable since it forces the researcher to think about effect size: here the alternative parameter does not necessarily represent what the population parameter must be if the test rejects the null hypothesis; rather, it represents such an effect that would be cared about within the context of the study. So, for the hypothesis test of:

$$H_0 : \theta = \theta_0 \text{ vs. } H_A : \theta = \theta_A$$

there are actually *three* quantities of interest: θ , θ_0 , and θ_A , where θ is the actual, unobserved value of the parameter of interest, and θ_0, θ_A are the researcher-determined point null and alternative values, where θ_A is determined

Algorithm 3. Monte Carlo Simulation

-
- 1: Generate a sample of size n from a normal distribution with mean μ , variance 1
 - 2: Perform a one-sample t-test of the hypothesis $H_0 : \mu = 0$
 - 3: Check whether the mean is greater than $\hat{\mu}_A/2$, where $\hat{\mu}_A$ is the “hypothesized” alternative mean. If so, reject H_0 .
 - 4: Repeat for all combinations of values of n, μ , and $\hat{\mu}_A$
-

by the effect the researcher is expecting to see. The simulation was structured as follows:

We tested for $n = 10, 20, 30, 50$, $\mu = 0, 0.3, 0.5, 0.7, 1$, and $\hat{\mu}_A = 0.3, 0.5, 0.7, 1, 1.5, 2, 2.5, 3, 3.5$. The entire process was repeated a total of $M = 10000$ times. Simulation was conducted using the statistical package R [13].

3 Results

As expected, simulation results showed that the Bi Error method generally outperformed the traditional NHST method in terms of accuracy, particularly when the hypothesized alternative mean was greater than the actual mean. Tabular results (see Tables 1, 2, 3, 4, 5, 6, 7 and 8) are available in Appendix A.

In order to simplify step 3 for researchers and students, we also provide a generalized derivation of the Youden Index. Suppose we are testing:

$$H_0 : \theta * \theta_0 \text{ vs}$$

$$H_A : \theta > \theta_0, \text{ where } \theta \text{ is an unknown parameter and } \theta_0 \text{ is a value.}$$

We restrict our attention to this case for simplicity, since the extension of our results to the case where $\theta < \theta_0$ and the two-tailed case is trivial.

Theorem 1. *Suppose f_0, f_A are the sampling distributions of a statistic X under the null and alternative hypotheses, respectively, such that f_0, f_A are symmetric and satisfy $|\mu - x_1| > |\mu - x_2| \implies f(|\mu - x_1|) < f(|\mu - x_2|)$ (that is, they decrease in the tails). Then $J = (\mu_0 + \mu_A)/2$.*

This result has been previously shown for the Normal distribution, which is a special case [3, 10, 15]. However, Theorem 1 is more general in that it applies to a wide range of distributions, including the case where the mean and/or variance do not exist.

Before presenting the proof, it is convenient to introduce the following definition:

Definition 1. *The Bi Error is defined as $\zeta(c) = \alpha + \beta$*

We now present the proof of Theorem 1.

Proof. Note, it suffices to minimize $2\zeta := \zeta_2 = 1 - F_0(x) + F_A(x)$. Assume, without loss of generality, that $\mu_A \sim \mu_0$. In this case, a critical value less than μ_0

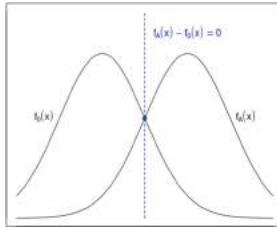


Fig. 2. The stationary point is found at $f_0(x) = f_A(x)$.

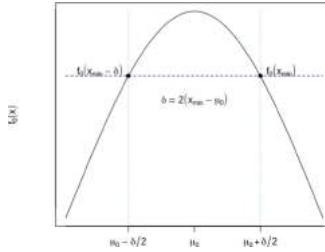


Fig. 3. Heuristic justification for $\delta = 2(x - \mu_0)$.

is not useful, so we will restrict our attention to $x > \mu_0$. So, $\frac{\partial \zeta_2}{\partial x} = -f_0(x) + f_A(x)$. Setting this equal to zero yields $f_0(x) = f_A(x)$ (see Fig. 2).

Since f_0, f_A are location family, this is equivalent to saying $f_0(x) = f_0(x - \delta)$, where $\delta = \mu_A - \mu_0$. But f_0 is unimodal. So the mode must be in the interval $(x - \delta, x)$. Also, since f_0 is symmetric, $\delta = 2(x - \mu_0)$.

$$\begin{aligned} \implies \mu_A - \mu_0 &= 2(x - \mu_0) \\ \implies \frac{\mu_A + \mu_0}{2} &= x. \end{aligned}$$

A heuristic explanation for the first step is available in Fig. 3.

Since f_0 is univariate, we know that this stationary point must be a maximum or a minimum. In order to show it is a maximum, we will show that the derivative is negative in a range antecedent to the point $x_0 = \frac{\mu_A + \mu_0}{2}$, and positive in a range immediately before it.

Now we will show

$$\forall \epsilon > 0, \zeta_2^{\epsilon} \left(\frac{\mu_0 + \mu_A}{2} + \epsilon \right) > 0 \quad (1)$$

Consider $z^{\epsilon}(x_0 + \epsilon) = -f_0(\frac{3\mu_A - \mu_0}{2} + \epsilon) + f_A(\frac{3\mu_A - \mu_0}{2} + \epsilon)$. So, in order for (1) to hold, we need $|f_A(\frac{3\mu_A - \mu_0}{2} + \epsilon)| < |f_A(\frac{\mu_0 + \mu_A}{2} + \epsilon)|$. But since f_A is symmetric and decreases in the tails, this holds if $\frac{3\mu_A - \mu_0}{2} \sim \frac{\mu_0 + \mu_A}{2} + 2(\mu_A - \frac{\mu_0 + \mu_A}{2})$. But the term on the right of that equation simplifies to $\frac{3\mu_A - \mu_0}{2}$ as required. An identical argument exists for $f^{\epsilon}(z - \epsilon) < 0$.

4 Discussion

4.1 On the Choice of Hypothesized δ

One glaringly obvious possible criticism of this method lies in the necessary choice of “alternative” distribution ($\hat{\mu}_A$, or equivalently $\delta = \mu_A - \mu_0$). Furthermore, it does not make sense to directly estimate this parameter from the data, (using it twice: once to estimate effect size, and then again to perform the test) since in the case of a symmetric distribution, if the null hypothesis is *actually* true, this will cause us to use a critical value of approximately $(\mu_0 + \hat{\mu}_A)/2 = 2\mu_0/2 = \mu_0$, which is obviously not useful, since it causes a Type I error of $\alpha = .5$.

In light of this, $\hat{\mu}_A$ should be, rather, chosen by the researcher in light of the context of the study (i.e. δ should be what effect, if one exists, that they are expecting). Furthermore, the estimate of μ_A should be identified *before* the analyst explores the data.

Note, if $\hat{\mu}_A$ is too close to μ_0 , this threatens the experiment with inconclusive results due to an unreasonably high Type I error. Similarly, if $\hat{\mu}_A$ is too far from μ_0 , this threatens the experiment with inconclusive results due to an unreasonably high Type II error.

The simulated results indicate that the method performs particularly well when the researcher marginally *overestimates* the expected effect, aka when $\mu_A > \mu$. We believe this lends additional credence to the validity of our method due to the fact that it appears to take advantage of a cognitive bias, but we defer a thorough discussion of this to a subsequent paper.

4.2 Normal Distribution

We have shown that the Bi Error is minimized when we choose $(\mu_0 + \mu_A)/2$ as a critical value. A natural question is, what difference does this make, as compared with conventional methods? That is, what does the Bi Error look like when we choose a critical value corresponding to $\alpha = 0.05$?

As suggested in the introduction, consider the case where f_0, f_A are normal distributions, with different location parameters. Clearly the normal distribution satisfies the conditions necessary for Theorem 1 to hold. We are interested in the change in Bi Error due to choosing α, β such that ζ is minimized, as opposed to choosing $\alpha = 0.05$. More formally, we want to look at $\xi = \zeta(z_{0.05}) - \zeta((\mu_0 + \mu_A)/2)$, where $z_{0.05}$ is the 95th percentile for $N(\mu_0, \sigma^2)$.

Plotting with fixed σ^2 revealed that ξ has a positive relationship with $\delta = \mu_A - \mu_0$, as shown in Fig. 4. Plotting with fixed μ_A, μ_0 and letting σ^2 vary revealed an inverse relationship between ξ and δ (Fig. 5). So, by definition, ξ increases with effect size.

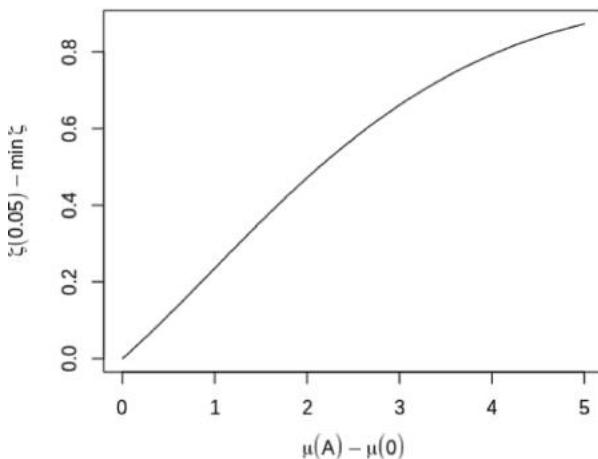


Fig. 4. Difference in ζ by $\mu_A - \mu_0$, with fixed $\sigma^2 = 2$

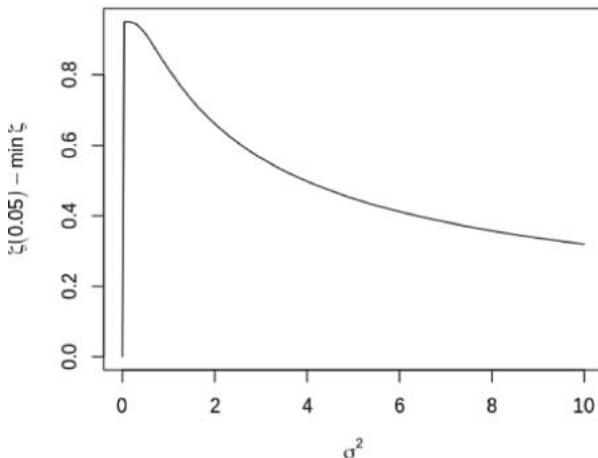


Fig. 5. Difference in ζ by σ^2 , with fixed $\mu_A - \mu_0 = 3$

Since ζ is a sum of probabilities, it is bounded. Also, the raw difference may have different meanings in different scenarios, depending on the actual values of ζ . So, it makes more sense to look at the odds ratio, ω_1/ω_2 , where $\omega_1 = \zeta(z_{0.05}/(1 - \zeta(z_{0.05})))$, $\omega_2 = \zeta(0.5(\mu_0 + \mu_A))/(1 - \zeta(0.5(\mu_0 + \mu_A)))$. Plotting $\phi_\zeta = \omega_1/\omega_2$ with effect size (Cohen's D), yields Fig. 6 wherein it is notable that, even for small effect sizes, the odds of making an incorrect conclusion on a hypothesis test are 4–6 times larger if we set $\alpha = 0.05$, rather than minimizing the Bi Error function.

This phenomenon is explained in greater depth when we consider the actual algebraic representation of ζ . Define ζ_α to be the Bi Error for a standard z test

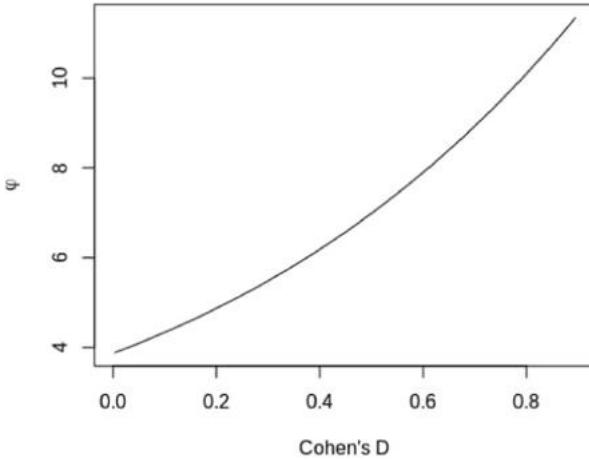


Fig. 6. Cohen's D vs ϕ , with $\sigma^2 = 5$, $\mu_0 = 0$, and μ_A ranges from 0 to 2

using the usual critical value x_α , chosen to achieve some fixed Type I error α . Similarly, define ζ^* to be the Bi Error for the same scenario, but using Theorem 1, which states that the minimum is found at $x^* = (\mu_0 + \mu_A)/2$. Then ξ is given by:

$$\begin{aligned}
\xi &= \zeta_\alpha - \zeta^* = \alpha + F_A(x_\alpha) + F_A(x_\alpha) - [1 - F_0(x^*) + F_A(x^*)] \\
&= \alpha + F_A(x_\alpha) + F_0(x^*) - F_A(x^*) \\
&= \alpha + [1 + 0.5erf((x_\alpha - x_0)/(\sigma\sqrt{2})) + 0.5erf(\frac{x^* - \mu_0}{\sqrt{2}\sigma}) - 0.5erf((x^* - \mu_A)/(\sqrt{2}\sigma))] \\
&= \alpha + 1 + 0.5erf((x_\alpha - \mu_0)/(2\sqrt{2}\sigma)) + 0.5erf((\mu_A - \mu_0)/(2\sqrt{2}\sigma)) - 0.5erf((\mu_0 - \mu_A)/(2\sqrt{2}\sigma)),
\end{aligned}$$

which can be rewritten in terms of Cohen's d, $d = (\mu_A - \mu_0)/\sigma$, like so:

$$\xi = \alpha + 1 + \frac{1}{2}erf((x_\alpha - \mu_0)/(2\sqrt{2}\sigma)) + \frac{1}{2}erf(d/(2\sqrt{2})) - \frac{1}{2}erf(-d/(2\sqrt{2}))$$

Thus, since the error function erf is known to be monotonically increasing, the last two terms of ξ must be increasing in d . Of course, this naturally leaves the question of the term $\frac{1}{2}erf((x_\alpha - \mu_0)/(2\sqrt{2}\sigma))$, which appears to be *decreasing* in d , since x_α is a monotonically increasing function of μ_0 . However, notice that since we have restricted ourselves to the single-tailed case, $x_\alpha > \mu_0$. So for any value of d , $\frac{1}{2}erf((x_\alpha - \mu_0)/(2\sqrt{2}\sigma)) < \frac{1}{2}erf((\mu_0 - \mu_A)/(2\sqrt{2}\sigma))$ and the last term sort of “absorbs” the one that is decreasing in d . Hence, for any small increase in d , ξ will also increase. Note, in the lower tailed scenario, this would be increasing in $-d$.

4.3 F Distribution

A natural additional inquiry is that of the performance of our critical value when the test statistic, under the null and alternative hypotheses, follows some kind of an F distribution. This is obviously critical for many results in regression analysis, ANOVA, and any test wherein we use the extra sum of squares principle.

Clearly, since the F distribution is not symmetric, it does not satisfy the conditions of Theorem 1, so we cannot use the closed-form solution for the critical value derived therein. Rather, for the purposes of this study, we instead provide specific examples for a few different choices of numerator and denominator degrees of freedom, and non-centrality parameter on the alternative sampling distribution. All of the minimizations were done numerically using the R function `optim()` [13].

First, we investigate null sampling distribution $f_0 F_{10,10,0}$ and alternative sampling distribution $f_A F_{10,10,10}$. Using the typical method of rejection, that is, $\alpha = .05$, yields a staggering $\zeta = .78$. That is, if the true non-centrality parameter is 10 and we have 10 numerator and denominator degrees of freedom, we have a 78% chance of coming to an incorrect conclusion! Dichotomizing using a critical value that minimizes ζ reduces this chance to $\zeta = .570$, which is still objectively outrageous, but a dramatic improvement from the above. As can be expected, the probability of a Type I error here is substantially larger than .05, at $\alpha = 0.3$. However, the probability of a Type II error is wildly reduced from $\beta_{\alpha=.05} = .732$ to $\beta_\zeta = .263$, further illustrating the necessity of a redefined standard for a critical value, in situations where Type I and Type II errors are equally egregious.

Changing the numerator and denominator degrees of freedom to 2 and 30, as is more common in practice, makes things better for both methods. The standard method ($\alpha = .05$) yields a Bi Error of 0.278, and probability of Type II error $\beta_{0.05} = 0.228$. Using a critical value that minimizes the Bi Error reduces the Bi Error to $\zeta = 0.236$, with $P(\text{Type I Error})$ of $\alpha_\zeta = 0.112$ and a $P(\text{Type II Error})$ of $\beta_\zeta = 0.123$. Notice in this case as well, even without the subjective error analysis part of the proposed method, critical value $\text{argmin}(\zeta)$ is much more appropriate than the classic critical value corresponding to $\alpha = .05$ for situations in which Type I and Type II error are equally egregious.

5 Conclusion

Misinterpretation of p -values and statistical significance testing remains a major problem in modern statistical practice. In order to eradicate the problem at its root, we have defined an accessible replacement for the existing, commonly-known 5-step hypothesis testing procedure. Our method possesses many desirable qualities, including but not limited to simplicity, lack of the need for p -values, potential for an inconclusive results outcome, and quantitative incorporation of Type II error. Theoretical and empirical results indicate that this method can be useful in many cases.

Notes and Comments. This method is meant as a replacement for the 5-step method for null hypothesis significance testing. It is meant primarily for students and researchers unfamiliar with statistics; as such its scope is limited with regards to real-world applications. One aspect of this is evident in the fact that Theorem 1 does not apply to F-distributions or the case where the variances of the null and alternative distributions are unequal. Further research is warranted in these areas.

Acknowledgment. We wish to thank Dr. Lynne Stokes and the two reviewers for their feedback, which substantially improved the paper. We additionally thank Dr. Dan Jeske for introducing us to the literature on hypothesis testing with neutral zones, as well as Michael Watts and Austin Marsteller for proofreading the manuscript.

Appendix

A Simulation Results

In order to assess the performance of each method, we calculated what proportion of the time H_0 was rejected, for each value of n, μ , and μ_A . Note, for $\mu = 0$ this represents Type I error, and for $\mu \neq 0$ this proportion represents statistical power. Re-running with a different seed yielded deviances in the third decimal place, so the Monte Carlo error on results is $\pm .01$. Consequently, results are reported to the second decimal place.

Table 1. Rejection rates using the Bi Error method, $n = 10$, given true location parameter (μ) and hypothesized alternative location parameter ($\hat{\mu}_A$)

μ (Actual)					
$\hat{\mu}_A \downarrow$	0	0.3	0.5	0.7	1
0.3	0.44	0.79	0.92	0.98	1.00
0.5	0.40	0.76	0.91	0.98	1.00
0.7	0.36	0.73	0.89	0.97	1.00
1	0.32	0.67	0.86	0.96	1.00
1.5	0.23	0.58	0.80	0.92	0.99
2	0.17	0.49	0.73	0.89	0.98
2.5	0.12	0.39	0.64	0.84	0.97
3	0.08	0.32	0.55	0.76	0.95
3.5	0.06	0.24	0.45	0.68	0.90

Vertical columns for the Bi Error method tables represent the hypothesized alternative mean, μ_A . All tables were generated using the R package [18].

Table 2. Rejection rates using NHST method with $\alpha = .05$, $n = 10$, given true alternative parameter

μ (Actual)					
	0	0.3	0.5	0.7	1
0.05	0.22	0.43	0.65	0.90	

Table 3. Rejection rates using the Bi Error method, $n = 20$, given true location parameter (μ) and hypothesized alternative location parameter ($\hat{\mu}_A$)

μ (Actual)					
$\hat{\mu}_A \downarrow$	0	0.3	0.5	0.7	1
0.3	0.44	0.89	0.98	1.00	1.00
0.5	0.40	0.86	0.98	1.00	1.00
0.7	0.36	0.84	0.97	1.00	1.00
1	0.31	0.81	0.96	1.00	1.00
1.5	0.22	0.72	0.93	0.99	1.00
2	0.17	0.63	0.89	0.98	1.00
2.5	0.11	0.54	0.84	0.97	1.00
3	0.07	0.45	0.77	0.94	1.00
3.5	0.05	0.35	0.69	0.91	1.00

Table 4. Rejection rates using the NHST method with $\alpha = .05$, $n = 20$, Given true alternative parameter

μ (Actual)					
	0	0.3	0.5	0.7	1
0.05	0.36	0.70	0.92	1.00	

Table 5. Rejection rates using the Bi Error method, $n = 30$, given true location parameter (μ) and hypothesized alternative location parameter ($\hat{\mu}_A$)

μ (Actual)					
$\hat{\mu}_A \downarrow$	0	0.3	0.5	0.7	1
0.3	0.45	0.93	1.00	1.00	1.00
0.5	0.40	0.92	0.99	1.00	1.00
0.7	0.37	0.90	0.99	1.00	1.00
1	0.31	0.87	0.99	1.00	1.00
1.5	0.23	0.82	0.98	1.00	1.00
2	0.16	0.74	0.96	1.00	1.00
2.5	0.11	0.66	0.93	0.99	1.00
3	0.07	0.56	0.89	0.99	1.00
3.5	0.04	0.46	0.84	0.98	1.00

Table 6. Rejection rates using the NHST method with $\alpha = .05$, $n = 30$, given true alternative parameter

μ (Actual)					
	0	0.3	0.5	0.7	1
0.05	0.48	0.85	0.98	1.00	

Table 7. Rejection rates using Bleile's critical value, $n = 50$, given true location parameter (μ) and hypothesized alternative location parameter ($\hat{\mu}_A$)

μ (Actual)					
$\hat{\mu}_A \downarrow$	0	0.3	0.5	0.7	1
0.3	0.45	0.98	1.00	1.00	1.00
0.5	0.41	0.97	1.00	1.00	1.00
0.7	0.37	0.96	1.00	1.00	1.00
1	0.32	0.95	1.00	1.00	1.00
1.5	0.23	0.92	1.00	1.00	1.00
2	0.17	0.87	1.00	1.00	1.00
2.5	0.11	0.81	0.99	1.00	1.00
3	0.07	0.74	0.98	1.00	1.00
3.5	0.04	0.65	0.96	1.00	1.00

Table 8. Rejection rates using the NHST method with $\alpha = .05$, $n = 50$, given true alternative parameter

μ (Actual)					
	0	0.3	0.5	0.7	1
0.05	0.67	0.97	1.00	1.00	

References

1. Beigel, J., et al.: Remdesivir for the treatment of COVID-19. *New England J. Med.* **383**, 1813–1826 (2020)
2. Cassidy, S.A., Dimova, R., Giguère, B., Spence, J.R., Stanley, D.J.: Failing grade: 89% of introduction-to- psychology textbooks that define or explain statistical significance do so incorrectly. *Adv. Methods Pract. Psychol.* **2**(3), 233–239 (2019)
3. Fluss, R., Faraggi, D., Reiser, B.: Estimation of the Youden index and its associated cutoff point. *Biom. J.* **47**(4), 129–133 (2005)
4. Harrell, F.: Statistical errors in the medical literature. <https://www.fharrell.com/post/ermed/>
5. Ioannidis, J.: What have we (not) learnt from millions of scientific papers with p -values? *Am. Stat.* **73**(Sup.1), 20–25 (2019)
6. Jeske, D.: Special collection on p -values. *Am. Stat.* **73**(Sup.1), 1–401 (2019)

7. Jeske, D.: Alternatives to the traditional p -value. National Institute of Statistical Science, Webinar (2019)
8. Jeske, D.: Digging deeper into p -values: Webinar follow-up with three authors. National Institute of Statistical Science, Webinar (2019)
9. Jeske, D., Smith, S.: Maximizing the usefulness of statistical classifiers for two populations with illustrative applications. *Stat. Methods Med. Res.* **27**, 1–15 (2016)
10. Li, C.: Partial Youden index and cut point selection. *J. Biopharmec. Stat.* **20**(5), 1520–5711 (2018)
11. Liu, X.: Classification accuracy and cut point selection. *Stat. Med.* **31**, 2676–2686 (2011)
12. Norrie, J.: Remdesivir for COVID-19: challenges of Underpowered studies. *Lancet* **395**, 1569 (2020)
13. R Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing (2013)
14. Senn, S.: Dichotomania: an obsessive-compulsive disorder that is badly affecting the quality of analysis in pharmaceutical trials. In: Proceedings of the International Statistical Institute, ISI Sydney, pp. 1–15 (2005)
15. Schistermann, E., Perkins, N.: Partial Youden index and cut point selection. *Commun. Stat.* **36**(3), 549–563 (2007)
16. Wasserstein, R., Lazar, N.: The ASA statement on p -values: context, process, and purpose. *Am. Stat.* **70**(2), 129–133 (2016)
17. Wasserstein, R., Lazar, N.: Moving towards a world beyond $p < .05$. *Am. Stat.* **73**(Sup.1), 1–19 (2019)
18. Dahl, D.B., Scott, D. Roosen, C., Magnusson, A., Swinton, J.: xtable: export tables to LaTeX or HTML, R Foundation for Statistical Computing, R package version 1.8-4 (2019)
19. Youden, W.J.: Index for rating diagnostic tests. *Cancer* **3**(1), 32–35 (1950)



A New Proposal of Parametric Similarity Measures with Application in Decision Making

Luca Anzilli¹(✉) and Silvio Giove²

¹ Department of Management, Economics, Mathematics and Statistics,
University of Salento, Lecce, Italy
luca.anzilli@unisalento.it

² Department of Economics, University Ca' Foscari of Venice, Venice, Italy
sgiove@unive.it

Abstract. We introduce a general equality index for two fuzzy values, proposing a parametric family of similarity measures between two fuzzy vectors, and investigating the mathematical properties. Subsequently we construct a class of parametric similarity measures, showing how the proposed approach extends and generalizes previously proposed framework in the context of decision making problem. The proposed approach can support decision maker under complex situations. An application of the proposed method is also given.

Keywords: Similarity measure · OWA operator · Decision making systems · Fuzzy systems

1 Introduction

Starting from the equality index originally proposed by Pedrycz [9], we propose an equality index among two fuzzy values, and use it to formalize a parametric family of similarity measures between two fuzzy vectors. Some properties of the new equality index are investigated, analyzing the relationships between similarity and aggregation operator in function of t-norm selection. In Sect. 2 we introduce some preliminary concept and formalize a generalization of the equality index proposed by Pedrycz. To this purpose, we consider some relationships between implication operator and t-norm. Section 3 particularizes these general results to the case of three commonly used t-norms, the probabilistic t-norm, the Dubois and Prade and the Schweizer and Sklar t-norms respectively. The main results are collected and proposed in the Proposition 3. The obtained results are used in Sect. 4 for a decision making problem originally proposed by [2, 3] where a particular class of OWA operators were proposed. The main idea is described in Subsect. 4.2, where depending on the choice of two parameters, many types of decision maker's behaviour can be represented and formalized. Finally, Sect. 5 presents some numerical simulation, and Sect. 6 reports some conclusions and suggestions for future work.

2 Preliminaries and Generalization of Equality Index

In [9, 10] Prof. Pedrycz introduced, in a logical framework, an equality index $q(x, y)$ for two fuzzy values $x, y \in [0, 1]$ as

$$q(x, y) = \frac{1}{2}(\mu_1(x, y) + \mu_2(x, y)) \quad (1)$$

where

$$\mu_1(x, y) = (x \rightarrow y) \wedge (y \rightarrow x)$$

describes the degree to which x implies y and vice versa, and

$$\mu_2(x, y) = \mu_1(\bar{x}, \bar{y}) = (\bar{x} \rightarrow \bar{y}) \wedge (\bar{y} \rightarrow \bar{x})$$

denotes a degree of similarity between complements $\bar{x} = 1 - x$ and $\bar{y} = 1 - y$. Here “ \rightarrow ” denotes an implication and \wedge is a t-norm.

We observe that the equality index $q : [0, 1]^2 \rightarrow [0, 1]$ is defined in (1) as the arithmetic mean of μ_1 and μ_2 .

The following result provides expressions for μ_1 , μ_2 and q when \rightarrow is the Lukasiewicz implication.

Proposition 1. *If \rightarrow is the Lukasiewicz implication defined by*

$$x \rightarrow y = \min(1, 1 + y - x), \quad (2)$$

then we have

$$\mu_1(x, y) = \mu_2(x, y) = 1 - |x - y| \quad (3)$$

and, moreover, the equality index defined in (1) is given by

$$q(x, y) = 1 - |x - y|. \quad (4)$$

Proof. First, we observe that Lukasiewicz implication satisfies the equality

$$x \rightarrow y = \bar{y} \rightarrow \bar{x} \quad (5)$$

and thus, from the symmetry property of t-norms, we have

$$(x \rightarrow y) \wedge (y \rightarrow x) = (\bar{x} \rightarrow \bar{y}) \wedge (\bar{y} \rightarrow \bar{x}).$$

So, we only have to show that $(x \rightarrow y) \wedge (y \rightarrow x) = 1 - |x - y|$. Denoting by T the t-norm \wedge and by S the dual t-conorm of T , defined by $S(a, b) = 1 - T(1-a, 1-b)$, we obtain

$$\begin{aligned} (x \rightarrow y) \wedge (y \rightarrow x) &= T(x \rightarrow y, y \rightarrow x) \\ &= T(\min(1, 1 + y - x), \min(1, 1 + x - y)) \\ &= T(1 + \min(0, y - x), 1 + \min(0, x - y)) \\ &= T(1 - \max(0, x - y), 1 - \max(0, y - x)) \\ &= 1 - S(\max(0, x - y), \max(0, y - x)) \\ &= 1 - |x - y|. \end{aligned}$$

□

Remark 1. We observe that in general indices μ_1 and μ_2 are not equals. For example, if we use as implication \rightarrow the Gödel implication defined as (see [6])

$$x \rightarrow y = \begin{cases} 1 & \text{if } x \leq y \\ y & \text{otherwise,} \end{cases}$$

we obtain $\mu_1(x, y) = x \wedge y$ and $\mu_2(x, y) = (1 - x) \wedge (1 - y)$ if $x \neq y$, and $\mu_1(x, y) = \mu_2(x, y) = 1$ if $x = y$. So, $\mu_1 \neq \mu_2$.

We observe that equality index given in (4) can be obtained by aggregating μ_1 and μ_2 using any idempotent operator, such as average, minimum and maximum operators, instead of the arithmetic mean as done in (1).

Our aim is to extend the notion of equality index given in (1) by considering a more general, not idempotent, aggregation between μ_1 and μ_2 , that is by defining

$$q(x, y) = \mathbb{A}(\mu_1(x, y), \mu_2(x, y)) \quad (6)$$

where $\mathbb{A} : [0, 1]^2 \rightarrow [0, 1]$ is an aggregation function [7]. In particular, in this study, we focus on the case when \mathbb{A} is a t -conorm S .

Definition 1. Given the two fuzzy values $x \in [0, 1]$ and $y \in [0, 1]$ we define

$$q(x, y) = S((x \rightarrow y) \wedge (y \rightarrow x), (\bar{x} \rightarrow \bar{y}) \wedge (\bar{y} \rightarrow \bar{x})) \quad (7)$$

where $\bar{x} = 1 - x$ is the complement of x , \wedge is a t -norm, \rightarrow denotes a (fuzzy) implication and S is a triangular conorm (t -conorm).

Remark 2. From the symmetry property of t -norms, the index q defined in (7) is symmetric, that is $q(x, y) = q(y, x)$.

We observe that if \rightarrow is the Lukasiewicz implication then, from (3), we get

$$q(x, y) = S(1 - |x - y|, 1 - |x - y|). \quad (8)$$

The choice of a t -conorm as aggregation function is motivated by the intention to introduce, following the above procedure, a less restrictive equality index than (4).

We observe that similarity indices can be built starting from a distance $d : [0, 1]^2 \rightarrow [0, 1]$ as

$$s(x, y) = 1 - d(x, y).$$

If the distance is $d(x, y) = |x - y|$, we retrieve the equality index (4), that can be also obtained from (8) using the t -conorm maximum (as it is idempotent) $S_M(x, y) = \max\{x, y\}$.

If the squared distance $d(x, y) = |x - y|^2$ is used, we obtain the less restrictive similarity index $s(x, y) = 1 - |x - y|^2$ that, as we will show in the next result, can be also obtained from (8) using the t -conorm probabilistic sum.

Proposition 2. If S is the t -conorm probabilistic sum¹ defined by

$$S_P(a, b) = 1 - (1 - a)(1 - b) = a + b - ab \quad (9)$$

then the equality index q defined in (7) computed using Lukasiewicz implication is given by

$$q = 1 - |x - y|^2. \quad (10)$$

Proof. From (8) and using (9) we get $q = S_P(1 - |x - y|, 1 - |x - y|) = 1 - |x - y|^2$. \square

So we have found that either the equality index proposed by Pedrycz $q = 1 - |x - y|$ or the equality index $q = 1 - |x - y|^2$ can be achieved in a logical framework using disjunctive operators, such as the t -conorm maximum S_M and the t -conorm probabilistic sum S_P , respectively.

3 Two Classes of Parametric Equality Indices

Starting from the results achieved in previous section and moving in the same logical framework, we will build two parametric classes of equality indices between $q = 1 - |x - y|$ and $q = 1 - |x - y|^2$. For this, we will use two parametric families of t -conorms.

Our first proposal, based on the family of t -conorms proposed by Dubois and Prade [4, 5], is presented in the next Proposition.

Proposition 3. If we employ the family of t -conorms proposed by Dubois and Prade [4, 5]

$$S_\alpha^{(1)}(a, b) = 1 - \frac{(1 - a)(1 - b)}{\max((1 - a), (1 - b), \alpha)} \quad \alpha \in]0, 1] \quad (11)$$

then the equality index defined in (7) using Lukasiewicz implication can be expressed as

$$q^{(1)} = 1 - \frac{|x - y|^2}{\max(|x - y|, \alpha)}. \quad (12)$$

Proof. From (8) and using (11) we obtain

$$q^{(1)} = S_\alpha^{(1)}(1 - |x - y|, 1 - |x - y|) = 1 - \frac{|x - y|^2}{\max(|x - y|, \alpha)}.$$

\square

Our second proposal, based on the family of t -conorms proposed by Schweizer and Sklar [1, 8] (with parameter α restricted in $]0, 1]$), is given in the next Proposition.

¹ The probabilistic sum is the dual t -conorm of t -norm product $T_P(a, b) = ab$.

Proposition 4. *If we employ the family of t-conorms proposed by Schweizer and Sklar [1, 8]*

$$S_{\alpha}^{(2)}(a, b) = (a^{1/\alpha} + b^{1/\alpha} - a^{1/\alpha}b^{1/\alpha})^{\alpha} \quad \alpha \in]0, 1] \quad (13)$$

then the equality index defined in (7) using Lukasiewicz implication is given by

$$q^{(2)} = (1 - |x - y|) \left[2 - (1 - |x - y|)^{1/\alpha} \right]^{\alpha}. \quad (14)$$

Proof. From (8) and using (13) we get

$$\begin{aligned} q^{(2)} &= S_{\alpha}^{(2)}(1 - |x - y|, 1 - |x - y|) = \left[2(1 - |x - y|)^{1/\alpha} - (1 - |x - y|)^{2/\alpha} \right]^{\alpha} \\ &= (1 - |x - y|) \left[2 - (1 - |x - y|)^{1/\alpha} \right]^{\alpha}. \end{aligned}$$

□

We observe that the parametric families of t-conorms (11) and (13) satisfy the following properties: if $\alpha = 1$ then $S_{\alpha}^{(1)}(a, b) = S_{\alpha}^{(2)}(a, b) = S_P(a, b)$; if $\alpha \rightarrow 0$ then $S_{\alpha}^{(1)}(a, b) \rightarrow S_M(a, b)$ and $S_{\alpha}^{(2)}(a, b) \rightarrow S_M(a, b)$.

In the following result we give some properties for equality indices $q^{(1)}$ and $q^{(2)}$ introduced in (12) and (14), respectively (we omit the trivial proof).

Proposition 5. *The following properties hold:*

- (i) *if $x = y$ then $q^{(1)} = q^{(2)} = 1$;*
- (ii) *if $|x - y| = 1$ then $q^{(1)} = q^{(2)} = 0$;*
- (iii) *if $\alpha = 1$ then $q^{(1)} = q^{(2)} = 1 - |x - y|^2$;*
- (iv) *if $\alpha = 0$ (the limit case) then $q^{(1)} = q^{(2)} = 1 - |x - y|$.*

In Fig. 1 we show the performance of equality indices $q^{(1)}$ and $q^{(2)}$ plotted as functions of fuzzy value x , with y fixed, for different values of parameter α .

Remark 3. If we express equality indices as functions of distance $d = |x - y|$ we have $q^{(1)} = 1 - \frac{d^2}{\max(d, \alpha)}$ and $q^{(2)} = (1 - d) \left[2 - (1 - d)^{1/\alpha} \right]^{\alpha}$. In Fig. 2 we have plotted equality indices $q^{(1)}$ and $q^{(2)}$ as functions of distance d for different values of parameter α .

We observe that for $d < \alpha$ the equality index $q^{(1)}$ agrees with the second order Taylor polynomial of $q^{(2)}$ centered at $d_0 = 0$. Indeed, denoting $f(d) = q^{(2)}$ we have, by computation, $f(0) = 1$, $f'(0) = 0$, $f''(0) = -\frac{2}{\alpha}$ from which it follows that the second order Taylor polynomial of $q^{(2)}$ at $d_0 = 0$ is given by

$$P_2(d) = f(0) + f'(0)d + \frac{1}{2}f''(0)d^2 = 1 - \frac{d^2}{\alpha}$$

and thus $P_2(d) = q^{(1)}$ for $d < \alpha$. As a consequence we deduce that, for a fixed value of parameter α , the two equality indices are very close when the distance between x and y is small. Conversely, their behaviour is very different when the distance between x and y increases. In particular, as shown in Fig. 3 equality index $q^{(2)}$ is less restrictive than $q^{(1)}$.

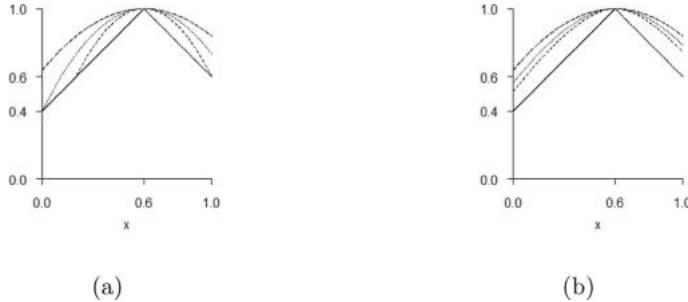


Fig. 1. Equality indices $q^{(1)}$ (a) and $q^{(2)}$ (b) Defined in (12) and (14) plotted as functions of x , with $y = 0.6$ fixed, for different values of parameter α : $\alpha = 0$ (continuous line), $\alpha = 0.4$ (dashed line), $\alpha = 0.6$ (dotted line) and $\alpha = 1$ (dot-dashed line).

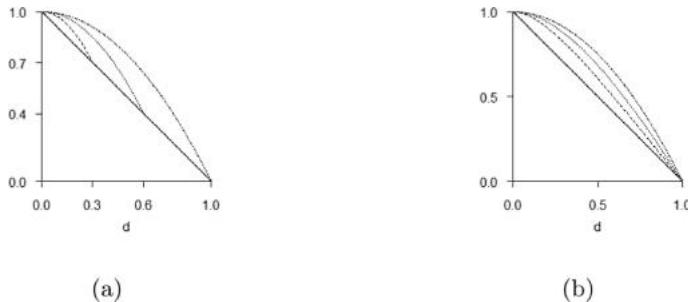


Fig. 2. Equality indices $q^{(1)}$ (a) and $q^{(2)}$ (b) plotted as functions of distance d for $\alpha = 0$ (continuous line), $\alpha = 0.3$ (dashed line) $\alpha = 0.6$ (dotted line) and $\alpha = 1$ (dot-dashed line).

4 Parametric Similarity Measures for Fuzzy Vectors

In this section we propose a similarity measure between two n -dimensional fuzzy vectors $X = (x_1, x_2, \dots, x_n) \in [0, 1]^n$ and $Y = (y_1, y_2, \dots, y_n) \in [0, 1]^n$ based on equality indices $q^{(1)}$ and $q^{(2)}$. To this aim we extend the idea presented in [2] using equality indices $q^{(1)}$ and $q^{(2)}$ instead of Pedrycz equality index. First we recall the definition of S-OWA operators, see [3].

4.1 Ordered Weighted Aggregation (OWA) Operators

Definition 2. An OWA operator of n dimension is a mapping $\text{OWA} : \mathbb{R}^n \rightarrow \mathbb{R}$ that has an associated weighting vector $W = (w_1, w_2, \dots, w_n)$ such that $w_i \in [0, 1]$ for all $i = 1, 2, \dots, n$ and $\sum_{i=1}^n w_i = 1$. Furthermore $\text{OWA}(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j b_j$ where b_j is the j -th largest among the a_i .

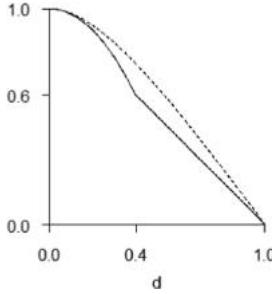


Fig. 3. Equality indices $q^{(1)}$ (continuous line) and $q^{(2)}$ (dashed line) plotted as functions of distance d for $\alpha = 0.4$.

The orlike S-OWA operator, denoted OWA_{SO} is defined by a family of OWA weights $W = (w_1, w_2, \dots, w_n)$ with $\alpha \in [0, 1]$ such that

$$w_i = \begin{cases} \frac{1}{n}(1 - \alpha) + \alpha & \text{for } i = 1 \\ \frac{1}{n}(1 - \alpha) & \text{for } i = 2, \dots, n. \end{cases}$$

We observe that

$$\text{OWA}_{\text{SO}}(a_1, a_2, \dots, a_n) = \alpha \text{Max}_i(a_i) + \frac{1}{n}(1 - \alpha) \sum_{i=1}^n a_i.$$

The andlike S-OWA operator, denoted OWA_{SA} is defined by a family of OWA weights $W = (w_1, w_2, \dots, w_n)$ with $\beta \in [0, 1]$ such that

$$w_i = \begin{cases} \frac{1}{n}(1 - \beta) & \text{for } i = 1, \dots, n - 1 \\ \frac{1}{n}(1 - \beta) + \beta & \text{for } i = n. \end{cases}$$

We note that

$$\text{OWA}_{\text{SA}}(a_1, a_2, \dots, a_n) = \beta \text{Min}_i(a_i) + \frac{1}{n}(1 - \beta) \sum_{i=1}^n a_i.$$

In [3] the following operator is defined

$$\text{OWA}_\lambda = \begin{cases} \text{OWA}_{\text{SO}} \text{ with } \alpha = 2\gamma - 1 & \text{if } \gamma \geq 0.5 \\ \text{OWA}_{\text{SA}} \text{ with } \beta = 1 - 2\gamma & \text{if } \gamma < 0.5. \end{cases} \quad (15)$$

We observe that the parameter γ is the value of orness of operator OWA_λ . The operator OWA_λ satisfies the following properties:

- (i) if $\gamma = 1$, that is $W = (1, 0, \dots, 0)$, then $\text{OWA}_\lambda = \text{Max}$;
- (ii) if $\gamma = 0$, that is $W = (0, \dots, 0, 1)$, then $\text{OWA}_\lambda = \text{Min}$;
- (iii) if $\gamma = 0.5$, that is $W = (1/n, , 1/n, \dots, 1/n)$, then $\text{OWA}_\lambda = \frac{1}{n} \sum_{i=1}^n a_i$.

4.2 A New Proposal

We now introduce two new similarity measures between fuzzy vectors using the equality indices defined in (12) and (14).

Definition 3. We introduce two parametric similarity measures for two fuzzy vectors $X = (x_1, x_2, \dots, x_n) \in [0, 1]^n$ and $Y = (y_1, y_2, \dots, y_n) \in [0, 1]^n$ defined, respectively, as

$$\text{Sim}_{\lambda, \alpha}^{(1)} = (X \equiv Y) = \text{OWA}_\lambda(q_1^{(1)}, q_2^{(1)}, \dots, q_n^{(1)}) \quad (16)$$

and

$$\text{Sim}_{\lambda, \alpha}^{(2)} = (X \equiv Y) = \text{OWA}_\lambda(q_1^{(2)}, q_2^{(2)}, \dots, q_n^{(2)}) \quad (17)$$

where $q_i^{(1)}$ is the equality index between x_i and y_i given in (12), $q_i^{(2)}$ is the equality index between x_i and y_i given in (14) and OWA_λ is the operator defined in (15).

Remark 4. We observe that, as special cases, for $\alpha = 0$ we retrieve the similarity measure proposed in [3] using Pedrycz equality index

$$\text{Sim}_\lambda = (X \equiv Y) = \text{OWA}_\lambda(q_1, q_2, \dots, q_n)$$

and for $\alpha = 0$ and $\gamma = 1/2$ we retrieve the definition given in [2] using Pedrycz equality index and arithmetic average as aggregation function $\text{Sim} = (X \equiv Y) = \frac{1}{n} \sum_{i=1}^n q_i$.

Remark 5. In the proposed approach the measurement of similarity between two fuzzy vectors $X = (x_1, x_2, \dots, x_n) \in [0, 1]^n$ and $Y = (y_1, y_2, \dots, y_n) \in [0, 1]^n$ is done in two steps: first, computing for each i the similarity of x_i and y_i , and then aggregating these similarities into a single similarity score. The choice between $\text{Sim}^{(1)}$ or $\text{Sim}^{(2)}$ and the selection of parameter α are related to the first step. The selection of parameter γ has influence on the second step. In particular:

- the choice between $\text{Sim}^{(1)}$ or $\text{Sim}^{(2)}$ is guided by the desire to be more or less restrictive, respectively, when fuzzy values x_i and y_i are not close to each other; in fact, as observed in Remark 3, in this case equality index $q^{(2)}$ is less restrictive than $q^{(1)}$;
- parameter α has effect on the function employed to measure the distance between x_i and y_i : for $\alpha = 0$ the distance is $d = |x_i - y_i|$, for $\alpha = 1$ we consider the squared distance is $d^2 = |x_i - y_i|^2$, when α goes from 0 to 1 the corresponding method of measurement similarity is less restrictive;

- parameter γ reflects the disposition of a decision maker about criteria: varying parameter γ from 0 to 1 we move from a pessimistic aggregation (logic ‘and’) to an optimistic perspective (logical ‘or’).

By a suitably tuning of the parameters, the user may combine different effects and, in particular, he can relax the concept of similarity according to his subjective opinion. The joint effect on similarity measure of parameters γ and α is illustrated in Fig. 4. The region (a) corresponds to a more restrictive requests for similarity degree either to criteria or to distance, the region (b) is less restrictive on criteria, the region (c) is less restrictive on distance, the region (d) corresponds to a less restrictive request either on criteria or on distance. The region (e) corresponds to an intermediate choice.

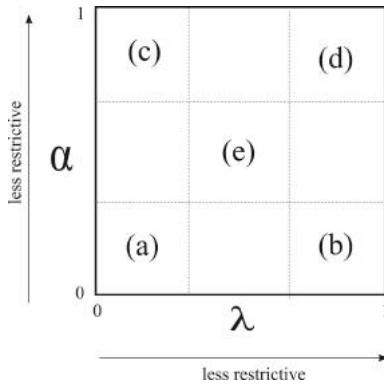


Fig. 4. Decision making table

5 Application

As illustrative example we apply methodology to an industrial inspection process of production quality presented in [3] with the only difference that we consider only one output. In this example a production quality process is expressed by a fuzzy inference system with six input variables x_1, \dots, x_6 (quality indices, criteria) and a single output y . Every production rule R_i , $i = 1, \dots, 5$, has the form

$$R_i : \text{ IF } x_1 \text{ is } X_{i,1} \text{ and } \dots \text{ and } x_6 \text{ is } X_{i,6} \text{ THEN } y \text{ is } Y_i$$

where $X_i = [X_{i,1}, \dots, X_{i,6}] \in [0, 1]^6$ is the fuzzy input data vector and $Y_i \in [0, 1]$ is the fuzzy output of rule R_i . For each rule R_i a parameter $c_i \in [0, 1]$ representing the belief of the rule and a threshold value $\delta_i \in [0, 1]$ (the rule is executed if the degree of matching is not less than δ_i) are given. Suppose that the fuzzy production system is given by

$$\begin{aligned}
X_1 &= [0.32 \ 1.00 \ 0.52 \ 0.25 \ 0.12 \ 0.95], & Y_1 &= 1.00, & c_1 &= 0.92, & \delta_1 &= 0.30 \\
X_2 &= [0.19 \ 0.92 \ 1.00 \ 0.51 \ 0.43 \ 0.37], & Y_2 &= 0.45, & c_2 &= 0.98, & \delta_2 &= 0.25 \\
X_3 &= [0.53 \ 0.74 \ 0.21 \ 1.00 \ 0.12 \ 0.64], & Y_3 &= 0.25, & c_3 &= 0.95, & \delta_3 &= 0.21 \\
X_4 &= [1.00 \ 0.73 \ 0.52 \ 0.33 \ 1.00 \ 0.72], & Y_4 &= 0.10, & c_4 &= 0.88, & \delta_4 &= 0.33 \\
X_5 &= [0.32 \ 0.62 \ 0.83 \ 0.71 \ 0.12 \ 1.00], & Y_5 &= 0.65, & c_5 &= 0.92, & \delta_5 &= 0.22 .
\end{aligned}$$

For an obtained feature vector $X = [x_1, x_2, x_3, x_4, x_5, x_6]$ we compute for each $i = 1, \dots, 5$ the similarity degree with fuzzy vector X_i by $s_i = Sim_{\lambda, \alpha}^{(k)} = (X \equiv X_i) = OWA_\lambda(q_{i,1}^{(k)}, q_{i,2}^{(k)}, \dots, q_{i,6}^{(k)})$ where $q_{ij}^{(k)} = (x_j \equiv X_{ij})$ is the equality index defined in (12) if $k = 1$ or in (14) if $k = 2$. Then for every $i = 1, \dots, 5$ we compute the modified similarity $\bar{s}_i = c_i \cdot s_i$ reflecting the certainty factor and, furthermore, and the value z_i obtained taking into account the threshold level

$$z_i = \begin{cases} \bar{s}_i & \text{if } \bar{s}_i \geq \delta_i \\ 0 & \text{otherwise.} \end{cases}$$

Finally, a fuzzy output value y is obtained from the max-min composition as $y = \max_{1 \leq i \leq 5} \min(z_i, Y_i)$. In Fig. 5 we show the numerical results obtained setting $X = [1.00, 0.45, 0.23, 0.67, 0.75, 0.53]$.

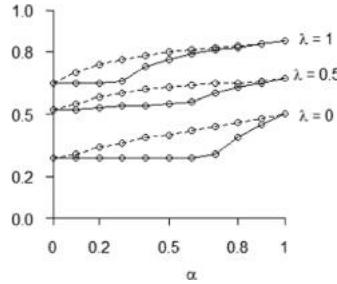


Fig. 5. Output y obtained using $Sim_{\alpha, \alpha}^{(1)}$ (continuous line) and $Sim_{\alpha, \alpha}^{(2)}$ (dashed line) by varying parameter α for different values of λ .

6 Conclusions

In this paper, following a logical framework, we have generalized the concept of equality index for two fuzzy values using aggregation operators. Furthermore, we have proposed parametric families of similarity measures, subsequently used to modify and generalize a decision support system based on a particular class of OWA operators. As confirmed by preliminary numerical simulations, the proposed tool is able to include different attitude of the definition, tuning the values of two real parameters. In a next step we intend to study a data driven procedure that starting from a set of samples (input and output variables) will optimize parameters' values.

References

1. Alsina, C., Schweizer, B., Frank, M.: *Associative Functions: Triangular Norms and Copulas*. World Scientific, Singapore (2006)
2. Bien, Z., Chun, M.-G.: An inference network for bidirectional approximate reasoning based on an equality measure. *IEEE Trans. Fuzzy Syst.* **2**(2), 177–180 (1994)
3. Chun, M.-G.: A similarity-based bidirectional approximate reasoning method for decision-making systems. *Fuzzy Sets Syst.* **117**(2), 269–278 (2001)
4. Dubois, D., Prade, H.: New results about properties and semantics of fuzzy set-theoretic operators. In: *Fuzzy Sets: Theory and Applications to Policy Analysis and Information Systems*, pp. 59–75. Plenum Press, New York (1980)
5. Dubois, D., Prade, H.: A class of fuzzy measures based on triangular norms a general framework for the combination of uncertain information. *Int. J. Gener. Syst.* **8**(1), 43–61 (1982)
6. Dubois, D., Prade, H.: Fuzzy sets in approximate reasoning, part 1: inference with possibility distributions. *Fuzzy Sets Syst.* **40**(1), 143–202 (1991)
7. Grabisch, M., Marichal, J.-L., Mesiar, R., Pap, E.: Aggregation functions: means. *Inf. Sci.* **181**(1), 1–22 (2011)
8. Klir, G.J., Yuan, B.: *Fuzzy Sets and Fuzzy Logic: Theory and Applications* (1996)
9. Pedrycz, W.: Direct and inverse problem in comparison of fuzzy data. *Fuzzy Sets Syst.* **34**(2), 223–235 (1990)
10. Pedrycz, W.: Neurocomputations in relational systems. *IEEE Trans. Pattern Anal. Mach. Intell.* **13**(3), 289–297 (1991)



The Challenges of Using Big Data in the Consumer Credit Sector

Kirill Romanyuk^(✉)

HSE University, Moscow, Russia

kromanyuk@hse.ru

Abstract. Credit risk analysis is essential in banking, and it relies on data. Multiple novel types of data are getting more popular for credit risk analysis in the consumer lending sector. On the one hand, the application of big data can increase the predictive power of credit risk analysis and reduce information asymmetry in the consumer lending market. On the other hand, more data types require better cyber-security, a more specific legal system for protecting consumers' privacy and promoting high standards of corporate ethics. This paper is focused on the challenges in the application of novel data in credit scoring. Concentrating big data including highly sensitive personal information in one place creates a high value target for hackers. For example, the credit bureau Equifax had a gigantic data breach in 2017, that exposed highly valuable information for more than half of adult US citizens. The data breach is described, including the potential value of private information that was compromised and the company's actions in terms of cyber-security and ethics, prior to and after the breach.

Keywords: Cyber security · Privacy · Ethics · Credit bureau

1 Introduction

An economy can be generally viewed as a flow of goods and services from sellers to buyers. However, a buyer can get a product that s/he did not really want due to a lack of information, which does not allow a buyer to assess the quality of this product properly, i.e. sellers usually possess more information about a product than buyers; this can be referred to as information asymmetry. The impossibility to fully observe outcomes and effort provision creates incentives for agents to act deceitfully [1]. Information asymmetry leads to the inefficiency of markets.

The credit market also suffers from information asymmetry. “In recent decades, a broad consensus has formed that most credit market failures are attributable to information asymmetries between lenders and borrowers” [2]. This may generally look like a one-way street that benefits borrowers, but it is not. Some borrowers do not qualify for a loan because of high credit risk. These borrowers have an incentive to hide credit-score lowering information. If they succeed, they can obtain a loan, but the portfolio of clients is now statistically

riskier in this bank. The higher the risk, the higher the loan rate; therefore, new credit in the next period will be more expensive for borrowers, other things being equal, and it has nothing to do with a banks capitalistic will to raise money, it is just mathematics applied to calculate the appropriate loan rate given this level of credit risk. A bank only tries to obtain its margin for transferring money from agents with excess funds to agents with a lack of money. Information asymmetry makes credit terms worse [3]. As a result, some borrowers with low creditworthiness try to benefit from borrowers with high creditworthiness by using information asymmetry to their advantage.

For many people, there is not enough information to evaluate credit risk (or credit score) by using standard methods. In the USA, 11% of the adult population lack credit records and 8.3% have credit records that are unscored [4]. These people usually face problems with having access to credit products; however, there is the reject inference problem [5], i.e. a bank does not know how rejected clients would perform, which makes it more difficult for the bank to optimize its portfolio of clients. Nevertheless, alternative data and methods to gather information and analyze it can be applied to make some of these people visible to banks. In addition, more information about clients leads to more precise credit risk assessment, and just a 1% improvement in precision can lead to significant profits. This work is focused on the challenges for the application of alternative data in credit scoring, including privacy, ethics and information asymmetry.

The rest of this paper is structured as follows. In the next section, the types of alternative data in the consumer credit sector are described, followed by explanation of the 2017 Equifax data breach. In Sect. 4, the potential value, challenges and solutions of information sharing for the economy are discussed, followed by concluding remarks.

2 Alternative Data in Credit Scoring

Creditworthiness assessment has been understood for centuries. Modern creditworthiness assessment (i.e. credit scoring) originated in the 1950s. It was based on a standard set of attributes but a lot more information can be gathered about a potential borrower today, and can be used for credit risk assessment. For instance, mobile phone usage can be analyzed to predict credit repayment [6]. The digital footprint (i.e. information that people leave online, simply by accessing or registering on a website) can be applied for predicting consumer default; moreover, easily accessible variables from the digital footprint, equal or exceed the information content of credit bureau scores, showing the discriminatory power for unscorable customers very similar to that of scorable customers [7]. As a result, BigTech companies have an information advantage in credit assessment, relative to traditional credit bureaus [8].

The types of structured non-traditional data include [9]:

- data on payments (e.g. utilities, mobile phone, and certain other obligations like rental information, taxes, etc.);

- crowdfunding transactions, factoring, leasing and credit insurance;
- transactions from P2P lending platforms, invoice, accounts payable, sales volume, merchant transactional data, mobile/e-money, procurement, historical business cash flows, shipping history, bills of lading, and data from online accounting platforms;
- data associated with assets (movables and fixed);
- payment flows received by disadvantaged individuals (e.g. subsidies, pensions, domestic and cross-border remittances, etc.).

The types of unstructured non-traditional data include [9]:

- social media and internet usage;
- emails;
- text and messaging files;
- audio files;
- digital pictures and images;
- GPS data;
- mobile usage (how many calls to the same number, peak usage, etc.);
- other metadata;
- psychographic, psychometric and other non-financial behavioral data.

2.1 Psychometric Information

Psychometric information is getting more popular in credit scoring because borrowers who appear more trustworthy usually have better credit scores and lower default rates [10]. Research in psychology and behavioral genetics suggests trustworthy appearance and trustworthy behavior could be linked through common biological factors [11,12]. Since personality traits are related to financial behavior, psychological traits could be used to lower credit risk in scoring models [13]. A psychometric test can lower the risk of the loan portfolio, when used as a secondary screening mechanism for clients with a credit history; however, it can also allow lenders to offer credit to some clients without a credit history, who were rejected based on their traditional credit scores without increasing the risk of the portfolio [14].

2.2 Social Networks

The data from social media can also be applied for assessing whether or not it is a good idea to lend money to customers without a credit score [15]. Social networks have changed our lives and perception of communication dramatically. However, information systems, empowered by social networks, have changed businesses and our lives much more significantly than just by allowing networking virtually anywhere. LinkedIn provides a platform for job searches empowered through a social network and search engines, allowing users to extract valuable information quickly, that is extremely difficult to obtain otherwise. Airbnb provides a

platform for renting and you can see what property was used by your friends and how they rated this place. Carsharing is becoming more popular because it allows you to use a car only when you need it without owning one, and it creates a network of people that use such a service and are dependent on online information about the availability of cars. Crowdfunding platforms allow people to accumulate funds for financing projects without standard instruments for raising funds such as bonds or bank loans. As a result, standard, well known businesses can be delivered in a very different way using information technologies.

2.3 Peer-to-Peer Lending

Peer-to-peer lending is an example of the successful application of information technologies to create a platform for lending. The volume of peer-to-peer lending grew 84% per quarter between 2007 and 2014, and it is expected to reach \$1 trillion by 2025 [16]. However, in the view of the potential global financial crisis triggered by lockdowns in many countries, this milestone may take longer to achieve. As a result, banks need to develop different tools and techniques in order to become prepared for competition with alternative ways of funding such as peer-to-peer lending.

Social networks allow a lender to extract additional information about a potential borrower in order to decide whether it is a good idea to support a loan application. For example, your friend's credit score might be low but you know all the details about his/her situation and can assume that his/her creditworthiness is actually better than the one that is shown by his/her credit score. However, according to Freedman and Jin borrowers with social ties are more likely to have their loans funded and are more likely to pay late or to default [17]. This generally looks reasonable because for potential borrowers with low creditworthiness friends can be the only option to obtain credit. Peer-to-peer lending platforms can use additional information (soft information) that is not used in a standard credit score. More informative decisions should be more accurate. For instance, decisions at [Prosper.com](#) (a peer-to-peer lending platform) were much more accurate with the application of soft information rather than credit score alone [16]. Nevertheless, the decision to lend money should be based on the analysis of creditworthiness even with application of soft information rather than the pure belief that "my friend won't default".

3 Privacy and Ethics

"Personal data has become a new currency in our digital era" [18]; therefore, some companies hunt for customers through the web by violating their privacy. For instance, merely by browsing the web, hosts can identify you through your digital footprint, and your browsing history can be used to suggest products that you might like to buy in ads on web pages. This looks helpful, nevertheless, a company can extract your email through automatically filled forms and start

sending you unsolicited emails. Your mobile phone can be also discovered through the web and when you are a few clicks away from buying something from an online shop you can get a message with a personal discount from a different online shop. Even though it can be beneficial for you to use such an offer in some cases, it is definitely a violation of your privacy.

3.1 Equifax Data Breach

The usage of personal information for advertising does not look harmful. Nevertheless, there is more valuable personal information stored by companies. These companies can declare that they will use this information only for pre-specified purposes but they can be hacked. For instance, 64% of Americans had personally experienced a major data breach before the 2017 Equifax data breach [19]. More than half of adult US citizens were affected by the 2017 Equifax data breach alone. Further, it was extremely valuable information which can affect at least retirement programs, the victims' credit, and future voter registrations [20]. As a result, more than half of the US adult population was exposed to identity theft, i.e. "the unlawful use of another's identifying information for gain" [21]; "identity theft often manifests itself through fraudulent use of existing accounts (e.g., credit card, telephone, online and insurance), the opening of new accounts or credit lines in the victim's name, as well as non-financial crimes" [22].

A data breach can happen even if security is strong in a company. Equifax had a data breach in 2015, in which data for about 10 million clients was compromised [23]. In 2017, there was a series of moments in which the data breach could be stopped but the level of disregard of clients' privacy was outrageous, and "there's no excuse for a breach involving so much sensitive personal data caused by failing to fix a known vulnerability with a ready-to-use patch" [20]. Equifax did not learn their lesson in 2015, cyber security was not improved enough, and privacy was not taken seriously.

The reasons why Equifax had such a position on privacy can be explained as follows. There is the privacy paradox, i.e. "people claim they are concerned about their exposed data but may not take protective actions" [24, 25]. There is the concept "too big to fall" that works in the US financial sector [20]. Equifax could have expected that if the damage was small, then people may not take action, and if the damage was large the government would help.

A September, 2017 poll found that 72% of Americans were concerned with the data breach and Equifax's ability to manage private sensitive data in the future; in addition, 59% of Americans would remove their data from Equifax if possible [23]; nevertheless, just a small percentage of people took protective measures because they were unaware of available tools or avoided using them [22]. This shows that privacy and security choices are affected by incomplete and asymmetric information flows, bounded rationality, cognitive biases and behavioral biases [26].

3.2 Ethical Aspects of the Equifax Data Breach

Equifax took the following actions that many people would find unethical:

- “Equifax’s official site for the data breach provided inconsistent results when consumers checked if they were affected” [22].
- “Equifax tried to bundle the free credit monitoring they offered with a forced arbitration clause, so that consumers would have waived their right to sue Equifax in class-action lawsuits” [22].
- Equifax delayed communication with the public; moreover, the management sold a significant amount of company stocks before the public announcement [23].

The 2017 Equifax data breach should be analyzed to make changes in policies in order to stimulate companies to take clients’ privacy, ethics and cyber security more seriously.

4 Information Sharing

Even though data breaches can reveal private information, information sharing theoretically makes economies more efficient by reducing information asymmetry. In the case of the credit market, sharing information about a client (even with credit bureaus like Equifax) generally looks reasonable because more information should lead to better credit risk evaluations and it is effective at altering borrower behavior, countering moral hazard, and improving repayment rates [27]. However, banks can try to take advantage of information from their own clients and refuse to share it [28]. According to Bruhm et al. such a disincentive for information sharing is relevant for very large banks [29]. The idea behind this disincentive is that in multi-period lending relationship if a loan is repaid in time, a borrower’s profile is improved and a borrower can expect better credit terms in the future [30]. A bank can try to share only negative information about clients (i.e. missed payments and defaults); therefore, a bank knows that a borrower is fine, but other banks view him/her as a bad borrower because only negative information was shared, creating an incentive for a client to stay with his/her current bank.

The problem for cross-country information sharing is in differences in national laws and regulations. Policy makers and regulators should coordinate and collaborate at the international level to develop cross-border data sharing standards and cross-border information regulations [31]. In addition, “in an environment where everyone can contribute data, much of it not traditional credit data (such as e-commerce, mobile, social, trade, and the like), and many types of entities can use this data (not just banks or traditional non-bank lenders), market alternatives to traditional credit reporting services may be needed” [32].

4.1 Regulations

The application of credit scoring methods can be negatively affected by privacy regulations [33]. One of the priorities in Argentina's G20 Presidency agenda was "The use of alternative data for credit reporting" [34]. In terms of "Data Privacy, Consumer Protection and Cyber Security" there are the following problems [31]:

- inadequate data privacy laws;
- inadequate transparency and disclosure regimes;
- the potential for alternative data usage that results in unacceptable forms of discrimination;
- restrictive or lack of adequate consent laws/regulations;
- growth in cyber risks and their potential impacts on global financial systems.

Recommendations for policy makers are as follows [31]:

- alternative data should be collected and processed lawfully;
- cost efficient consent mechanisms, where necessary;
- the accuracy and reliability of alternative data;
- consumer access and the ability to correct their information as well as request data deletion, where appropriate. Consumers should also be able to object to the processing of their information and should be accorded an opportunity to transfer their data to any other service provider;
- cybersecurity risk assessments are embedded into the overall risk management policies and procedures of industry participants;
- industry participants implement clear processes that guarantee consumers receive all the relevant information about data collection;
- the use of alternative data does not unfairly discriminate against protected groups and that participants should adopt measures to ensure that the predictiveness of alternative data is tested and verified.

It should be noted that point 4 is double-edged for credit scoring, because if customers are allowed to delete their credit histories, then those clients with bad credit histories will actively use this option, making credit bureaus useless. All in all, one of the objectives should be a reduction of opportunities for taking advantage of information asymmetry.

4.2 Discrimination and Manipulation

Algorithmic decision-making systems are becoming more popular for preventing social inequalities (e.g. criminal justice, human resource management, social work, credit and insurance [35–38]). Generally, 4 processes are involved in algorithmic decision making, i.e. to prioritize, classify, associate, and filter [39], which can be expressed in terms of mathematics without mentioning any biases; nevertheless, discrimination and censoring can appear. Social groups with protected attributes (e.g. race) should not be discriminated against. For instance, black defendants are significantly more likely to be incorrectly classified as high risk

than white defendants [40]. Filters should not become a censoring instrument. For example, Facebook can affect voter turnout in elections based merely on the amount of hard news promoted in an individual's news feed, because the Facebook newsfeed is algorithmically curated, and it is a source of news about government and politics for more than a half of millennials [39]; therefore, politicians can have an incentive to pay Facebook in order to create a censoring instrument in the news section against their opponents. In this case, voters may have the feeling that they see the full picture while what they actually observe is a biased picture.

The last point in recommendation for policy makers (the use of alternative data does not unfairly discriminate against protected groups) is a statistical challenge. Alternative data allows the identification of people from a protected group with a high probability in some cases. For this purpose, there are algorithmic solutions for discrimination prevention, i.e. data preprocessing, model post-processing and model regularization [41]. On the other hand, was it an effective idea to "protect" some groups in the first place? For example, gender usage prohibition in credit scoring was aimed against discrimination against women. In practice, it works in favor of men because when gender is used in credit scoring, it statistically shows that women are more creditworthy than men when all other attributes (e.g. salary, property) are the same [42]. As a result, policies should be created based on real data analysis, otherwise, a beautiful declaration and the harsh reality can be far from each other. However, securing privacy and non-discrimination come with the cost of information loss (e.g. [43]); therefore, the objective is to minimize information loss while ensuring the desired level of privacy and fairness.

Another question is should banks try to fully automate credit decisioning in the consumer lending sector. In some cases, a bank can leave a lending decision for credit managers. For example, in the mortgage lending market loan amounts are relatively high, so more individualized decisions can be beneficial for banks. Data visualization can be useful to support more informative credit decisions.

5 Discussion

In this paper, the challenges for the application of novel data in credit scoring has been discussed. Novel data in credit changes the landscape of credit scoring dramatically. There are many more scorable clients now because clients without a credit history or with insufficient information for standard credit scoring can be assessed based on alternative data. A credit score based solely on alternative data can even outperform a credit score based only on standard information. Banks with access to alternative data have a significant advantage over banks without this. For example, clients with a high credit score based on standard and alternative data can be targeted from a bank that uses only standard information and evaluates his/her credit score lower, providing worse credit terms.

New data types should not be used by companies as a way to violate privacy and overcome restrictions on protected attributes, i.e. discriminate against

protected groups. This issue is mentioned in G20 official documents, indicating its international value. In practice, avoiding discrimination against protected groups can be a difficult task and it comes with the cost of information loss that can result in financial loss. Large storage of private data like credit bureaus are high value targets for hackers. Equifax is an example of such a high value target. When Equifax staff ignored vulnerabilities in their system and did not apply a ready-to-use patch, hackers quickly used this opportunity. If alternative data is collected in addition to standard data, then more damage can be done if hackers attack, and more vulnerabilities are possible in the system; therefore, cybersecurity should be at a very high level. The availability of new types of data in credit scoring provides information for a more precise analysis of credit risk and requires more efforts to protect clients' privacy.

Acknowledgments. Support from the Basic Research Program of the National Research University Higher School of Economics is gratefully acknowledged.

References

1. Hopp, C., Speil, A.: Estimating the extent of deceitful behaviour using crosswise elicitation models. *Appl. Econ. Lett.* **26**(5), 396–400 (2019)
2. Doblas-Madrid, A., Minetti, R.: Sharing information in the credit market: contract-level evidence from US firms. *J. Financ. Econ.* **109**(1), 198–223 (2013)
3. Crawford, G.S., Pavanini, N., Schivardi, F.: Asymmetric information and imperfect competition in lending markets. *Am. Econ. Rev.* **108**(7), 1659–1701 (2018)
4. Brevoort, K.P., Grimm, P., Kambara, M.: Credit invisibles and the unscored. *Cityscape* **18**(2), 9–34 (2016)
5. Anderson, B.: Using Bayesian networks to perform reject inference. *Expert Syst. Appl.* **137**, 349–356 (2019)
6. Bjorkgren, D., Grissen, D.: Behavior revealed in mobile phone usage predicts credit repayment. *The World Bank* (2019)
7. Berg, T., Burg, V., Gomovic, A., Puri, M.: On the rise of fintechs-credit scoring using digital footprints. *Natl. Bureau Econ. Res.* (w24551) (2018)
8. Frost, J., Gambacorta, L., Huang, Y., Shin, H.S., Zbinden, P.: BigTech and the changing structure of financial intermediation. *Econ. Policy* **34**(100), 761–799 (2020)
9. Barci, G., Andreeva, G., Bouyon, S.: Data sharing in credit markets: does comprehensiveness matter? *Eur. Credit Res. Inst.* **23** (2019)
10. Duarte, J., Siegel, S., Young, L.: Trust and credit: the role of appearance in peer-to-peer lending. *Rev. Financ. Stud.* **25**(8), 2455–2484 (2012)
11. Cesarini, D., Dawes, C.T., Fowler, J.H., Johannesson, M., Lichtenstein, P., Wallace, B.: Heritability of cooperative behavior in the trust game. *Proc. Natl. Acad. Sci.* **105**(10), 3721–3726 (2008)
12. Kogan, A., Saslow, L.R., Impett, E.A., Oveis, C., Keltner, D., Saturn, S.R.: Thin-slicing study of the oxytocin receptor (OXTR) gene and the evaluation and expression of the prosocial disposition. *Proc. Natl. Acad. Sci.* **108**(48), 19189–19192 (2011)
13. Liberati, C., Camillo, F.: Personal values and credit scoring: new insights in the financial prediction. *J. Oper. Res. Society* **69**(12), 1994–2005 (2018)

14. Arraiz, I., Bruhn, M., Stucchi, R.: Psychometrics as a tool to improve credit information. *World Bank Econ. Rev.* **30**, S67–S76 (2017)
15. Kulkarni, S.V., Dhage, S.N.: Advanced credit score calculation using social media and machine learning. *J. Intell. Fuzzy Syst.* **36**(3), 2373–2380 (2019)
16. Gao, Q., Lin, M., Sias, R.W.: Words matter: The role of texts in online credit markets. Available at SSRN 2446114 (2018)
17. Freedman, S., Jin, G.Z.: The information value of online social networks: lessons from peer- to-peer lending. *Int. J. Ind. Organ.* **51**, 185–222 (2017)
18. Roman, D., Stefano, G.: Towards a reference architecture for trusted data marketplaces: the credit scoring perspective. In: 2nd International Conference on Open and Big Data, pp. 95–101. IEEE, Vienna (2016)
19. Olmstead, K., Smith, A.: Americans and cybersecurity. Technical report, The Pew Research Center (2017)
20. Berghel, H.: Equifax and the latest round of identity theft roulette. *Computer* **50**(12), 72–76 (2017)
21. White, M.D., Fisher, C.: Assessing our knowledge of identity theft: the challenges to effective prevention and control efforts. *Crim. Justice Policy Rev.* **19**(1), 3–24 (2008)
22. Zou, Y., Mhaidli, A.H., McCall, A., Schaub, F.: “I’ve got nothing to lose”: consumers’ risk perceptions and protective actions after the equifax data breach. In: Fourteenth Symposium on Usable Privacy and Security, pp. 197–216. The Advanced Computing Systems Association, Baltimore (2018)
23. Novak, A.N., Vilceanu, M.O.: “The internet is not pleased”: Twitter and the 2017 Equifax data breach. *Commun. Rev.* **22**(3), 196–221 (2019)
24. Acquisti, A., Brandimarte, L., Loewenstein, G.: Privacy and human behavior in the age of information. *Science* **347**(6221), 509–514 (2015)
25. Kokolakis, S.: Privacy attitudes and privacy behaviour: a review of current research on the privacy paradox phenomenon. *Comput. Secur.* **64**, 122–134 (2017)
26. Acquisti, A., et al.: Nudges for privacy and security: understanding and assisting users’ choices online. *ACM Comput. Surv.* **50**(3), 1–44 (2017)
27. De Janvry, A., McIntosh, C., Sadoulet, E.: The supply-and demand-side impacts of credit market information. *J. Dev. Econ.* **93**(2), 173–188 (2010)
28. Pagano, M., Jappelli, T.: Information sharing in credit markets. *J. Financ.* **48**(5), 1693–1718 (1993)
29. Bruhn, M., Farazi, S., Kanz, M.: Bank competition, concentration, and credit reporting. The World Bank (2013)
30. Comeig, I., Fernandez-Blanco, M.O., Ramirez, F.: Information acquisition in SME’s relationship lending and the cost of loans. *J. Bus. Res.* **68**(7), 1650–1652 (2015)
31. Use of alternative data to enhance credit reporting to enable access to digital financial services by individuals and SMEs operation in the informal economy. G20 paper, Global Partnership for Financial Inclusion (2018)
32. Owens, J.V., Wilhelm, L.: Alternative data transforming SME finance (English). World Bank Group, Washington, D.C. (2017)
33. Oskarsdottir, M., Bravo, C., Sarraute, C., Vanthienen, J., Baesens, B.: The value of big data for credit scoring: enhancing financial inclusion using mobile phone data and social network analytics. *Appl. Soft Comput.* **74**, 26–39 (2019)
34. Argentina’s G20 presidency 2018 priorities paper. Global Partnership for Financial Inclusion (2018)
35. Chalfin, A., et al.: Productivity and selection of human capital with machine learning. *Am. Econ. Rev.* **106**(5), 124–127 (2016)

36. Noriega-Campero, A., Bakker, M.A., Garcia-Bulle, B., Pentland, A.S.: Active fairness in algorithmic decision making. In: 2019 AAAI/ACM Conference on AI, Ethics, and Society, pp. 77–83 (2019)
37. Gillingham, P.: Predictive risk modelling to prevent child maltreatment and other adverse outcomes for service users: inside the black box of machine learning. *Br. J. Soc. Work* **46**(4), 1044–1058 (2015)
38. Siegel, E.: Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die. Wiley, Hoboken (2013)
39. Diakopoulos, N.: Accountability in algorithmic decision making. *Commun. ACM* **59**(2), 56–62 (2016)
40. Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A.: Algorithmic decision making and the cost of fairness. In: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 797–806. Association for Computer Machinery, New York (2017)
41. Zliobaite, I.: Measuring discrimination in algorithmic decision making. *Data Min. Knowl. Disc.* **31**(4), 1060–1089 (2017)
42. Andreeva, G., Matuszyk, A.: The law of equal opportunities or unintended consequences?: the effect of unisex risk assessment in consumer credit. *J. R. Stat. Soc. A. Stat. Soc.* **182**(4), 1287–1311 (2019)
43. Hajian, S., Domingo-Ferrer, J., Farras, O.: Generalization-based privacy preservation and discrimination prevention in data publishing and mining. *Data Min. Knowl. Discov.* **28**(5–6), 1158–1188 (2014)



New Engine to Promote Big Data Industry Upgrade

Jing He^{1(✉)}, Chuyi Wang², and Haonan Chen³

¹ School of Journalism and Communication, Tsinghua University, Beijing, China
hejing19@tsinghua.org.cn

² School of Resources, Environment and Jewelry, Jiangxi Vocational College of Applied Technology, Ganzhou, China

³ College of Geoscience and Surveying Engineering,
China University of Mining and Technology-Beijing, Beijing, China

Abstract. In 2020, the development of the big data industry will face severe challenges of transformation and upgrading. This article explores a series of issues in the big data industry upgrade from two levels: big data industrialization and big data industry chain. For the industrialization of big data, the industry starts from the core construction of big data industrialization, utilizes general big data platform as framework, and aggregates the horizontal industrial ecological layout and the vertical industrial ecological chain to form horizontal and vertical big data platforms respectively. The industry chain for big data is equipped with a docking mechanism that is different from the traditional chain association relationship. Through the multi-dimensional support and expansion of the big data industry chain, the framework and map of the industry chain are constructed, and the generation of a variety of technical models, system and methods of the big data industry chain is promoted. Finally, a big data product system composed of hardware products, software products, and integration products of both is built. The above analysis will help improve the service performance of the big data industry, improve the data connectivity of technology and the environment, and ultimately promote the big data industry upgrade.

Keywords: Big data · Big data industrialization · Big data industrial chain · Industrial upgrade

1 Introduction

Big data is increasingly becoming an indispensable strategic resource for governments at all levels to develop local economy and major industries to seize technological opportunities. By 2030, China's total data will exceed 4YB, accounting for 30% of the world's total. As a core means of production, big data spans the natural sciences and social and humanities. Various countries have placed big data at the national level for strategic deployment. The China Development and Reform Commission, the Economic and Information Commission and other departments have successively issued a series of documents to promote the development of the big data industry. Large-capacity, high-speed,

and diversified big data assets have been formed, including environmental protection big data, financial big data, tourism big data, cultural big data, etc. [1]. How to apply these data more complex and effective? The transformation and upgrading of the big data industry is the only way. The rapid development of emerging technologies such as artificial intelligence, quantum computing, 5G, and the Internet of Things provides strong support for the transformation and upgrading of the big data industry. However, there are some problems in the transformation and upgrading of the big data industry, such as the difficulty of big data processing and analysis (Huge quantity, various types, etc.), high requirements for data platform tools (lagging iterations, solidification of modules, etc.) [2], immature development of the big data industry (low data openness, unbalanced regional development, etc.) [3], restricting the rate of transformation and upgrading of the big data industry.

2 Big Data Industry Upgrade and Construction Goals and Key Breakthroughs

The network space, information space, and physical world connected by the communication network, the Internet and the Internet of Things will surely bring huge growth space and market value. At the same time, industrial opportunities arise in time. The artificial intelligence industry has great potential, and artificial intelligence applications will usher in huge breakthroughs in the field of service robots and computer vision [4]; the Internet of Things industry is growing rapidly, and the Internet of Everything will promote the full blossoming of the industrial field; the development of the cloud computing industry is gradually mature, and the service computing, Container orchestration, cross-cloud management, blockchain and other technologies are sought after and become hotspots. The cloud computing industry structure is transformed with technical facilities as the core to services; virtual reality and mixed reality are still worth looking forward to, virtual touch, eye tracking, light field display. Other technologies will usher in greater development. The big data industry will eventually realize the interconnection of the industrial ecosystem, driven by the opportunities of related industries, and realize both quantitative accumulation and qualitative evolution in the process of connection evolution.

The transformation and upgrading of the big data industry requires new breakthroughs in terms of theoretical foundation, practical application, and service decision-making [5]. Because the data types of big data analysis are usually structured data, semi-structured data and unstructured data, it is very difficult to mine and analyze [6]. For example, for data mining algorithms, it needs to be based on different data types and formats to scientifically present the characteristics of the data itself, but when determining the model characteristics of multi-source data, it is difficult to ensure the accuracy, complexity, cost performance, and automation. Therefore, it is necessary to seek breakthroughs in the upgrade of the big data industry from the bottlenecks of visual analysis, data mining algorithms, predictive analysis capabilities, semantic engines, data quality, and data management.

- 1) A key breakthrough in data fusion applications. Fusion means the integration of information sources. If it is the fusion of similar data, the fusion starts directly from

the data layer, that is, the data layer fusion. If it is the fusion of different types of data, feature fusion or decision fusion can be used to shield the heterogeneity of the data itself. Big data fusion applications require certain related operations, so that data fusion applications can act as a bridge connecting various modules in the big data platform, or can generate new functional modules to overcome differentiation, improve the exchange mechanism, and ensure information security, optimize analysis logic, eliminate decision-making interference, etc.

- 2) A key breakthrough in data analysis and processing. Big data analysis and processing methods include big data collection with high concurrency, centralized big data import/preprocessing, statistics/analysis based on distributed databases or distributed computing clusters, data mining based on single thread, etc. According to the diverse needs of data analysis and processing and multiple types of feature dimensions, there are currently a variety of typical big data computing modes, big data computing systems, and big data computing tools, but in data association construction, comprehensive analysis, model training, and visualization. Many aspects such as technology and data management mechanism need to be further updated and iterated to adapt to different big data analysis and processing applications.
- 3) A key breakthrough in basic support technology. With the rapid development of the digital economy, fundamental support technologies are flourishing, and the development of new technologies such as cloud computing, 5G, and artificial intelligence has brought new opportunities for the upgrading of the big data industry, enabling a unified structure and unified management of data collection and storage. Big data storage efficiency, data access efficiency, and security performance in the cloud environment need to be improved. The large-scale deployment of 5G in the construction of wireless networks, bearer networks, core networks, and terminals needs to be improved. The Internet of Things and edge computing are constantly generating. It is difficult for data to form a unified real-time processing, batch processing specification and other issues, and it is still far from the road to complete information, digitization, and automation of operation, operation, management and control, and decision-making.
- 4) Hierarchical breakthroughs in data fusion, data analysis and processing, and basic support technologies are a major move to introduce power trains that have upgraded the big data industry into high-speed railways. For data fusion applications, it is necessary to realize intelligence with AI as the core, verticalization with innovative positioning as service tactics, and sceneization with ubiquitous data service flow as the form of experience; for data analysis and processing, real-time processing feedback is required. The high efficiency of massive data, real-time streaming data with enhanced processing capabilities, and cloud-based data processing for one-stop analysis services; for basic supporting technologies, it is necessary to realize the “enterprise + government” cooperation mode connectivity, information systems and public data. The openness and sharing of interconnection, the technical integration of “5G + artificial intelligence + cloud computing + Internet of Things + blockchain”. This level of breakthrough will solve the data control, mining and utilization problems brought about by the upgrade of the big data industry, and realize the rapid transformation of information to data, data to knowledge, knowledge to decision-making, and decision-making to profit.

3 Realization of Big Data Industrialization

The report of the 19th National Congress of the Communist Party of China pointed out the direction of industrial construction. The future development direction of my country's big data industry construction must meet the requirements of big data strategic transformation, and present new characteristics and development plans in terms of industrial core and operating mode. In the context of the new digital age, the industrial upgrading of big data is closely related to various factors such as the policy environment, market structure, and industrial ecology. To promote the industrialization of big data, it is necessary to optimize the macro policy environment, further intensify the market competition pattern, introduce new and subdivided products and services in vertical fields, upgrade the underlying basic technologies such as fog computing, data lakes and blockchain, and enable industrial applications to multiple industries, building an industrial ecology in harmony [7].

3.1 Industrialization Core Construction—"General Big Data Platform+" Model

From the perspective of industrialization, the "universal big data platform+" model uses the universal big data platform as the framework to aggregate the change data of the horizontal industrial ecological landscape and the vertical industrial ecological chain to form horizontal and vertical big data platforms respectively. For the "universal big data platform + horizontal" model, that is, data covering various industry segments are aggregated on the universal big data platform to construct a relatively complete industrial ecological map, and through continuous aggregation effects, various horizontal Deep-processed knowledge products across various industries provide intelligent services for industry sectors in various sub-sectors; for the "universal big data platform + vertical" model, the industry ecological chain is "parts company-product manufacturing company-same industry company-related The upstream and downstream related data of "industry companies" are gathered on the general big data platform, and the interaction between each link is analyzed and mined through data analysis and mining to generate deep-processed knowledge products of the industrial process chain, and provide intelligence for related enterprise departments in the ecosystem policy support.

The "Universal Big Data Platform+" model is used to build a big data platform and open up multiple types of databases to serve horizontal and vertical big data platform models. Building a big data platform includes a big data analysis system development platform, a big data analysis system testing and evaluation platform, a big data analysis visualization display platform, and a major application demonstration and system integration platform. Among them, the big data analysis system development platform is a support platform and system specifically for big data analysis technologies in various subdivisions. It realizes the efficient parallelization of big data machine learning algorithms and provides unified and user-transparent scheduling for various machine learning algorithms. Process; the big data analysis system test and evaluation platform can realize the subjective and objective quality testing and application verification of the big data analysis methods and systems in various sub-fields under multi-source data objects and application scenarios; the big data analysis visualization display platform provides advanced visualization display Environment, providing efficient visualization of the

effects of big data analysis in various horizontal and vertical fields; major application demonstration and system integration platforms are mainly built through the establishment of major big data applications including multiple applications related to upstream and downstream industries in the vertical field Demonstration and system integration. The above four big data platform modules all adopt service-oriented architecture (service oriented architecture, SOA) to solve data processing, sharing and service problems. The technical standards of the services are adjusted according to actual needs, and cloud service standards. You can use an internal unified approach and can be commercialized [7]. According to this, the big data platform is constructed in the following way.

- 1) Build an open and shared system architecture, strengthen sharing, integrate common and open applications. SOA is a method and model of software architecture design. From a business perspective, everything is based on maximizing the value of “service”, using various existing software systems in various fields, industries, and departments to re-integrate and build a set. Emerging software architecture that integrates existing services in real time. Under the technical framework of SOA, a large and disorderly system can be integrated into a comprehensive and orderly system to achieve maximum IT asset utilization.
- 2) Build an open and shared “resource pool”, including hardware resource pool and shared data resource pool. In view of the high energy consumption of the current information computer room infrastructure, the difficult operation and maintenance of information systems, and the insufficient utilization of information resources, the information system hardware resource pool is based on the design of server domains, network domains, storage domains, and security domains, using virtualization, Cloud computing, edge computing, blockchain and other technologies realize dynamic scheduling, on-demand distribution, shared utilization and unified management of information infrastructure to reduce equipment energy consumption and operating costs, while improving operating efficiency.
- 3) Build an open and shared data environment, strengthen investigation, evaluation and dynamic monitoring, promote data interconnection, and organize a multi-level, multi-node “big data network”, making the big data platform increasingly digital, refined, and scientific. The integrated operating environment has the characteristics of real-time data acquisition, efficient data fusion, intelligent data processing, and networked data services, including building data as a service (DaaS), infrastructure as a service (IaaS), and an open and shared resource environment of SaaS as a service (SaaS), knowledge as a service (KaaS) and platform as a service (PaaS) [7].
- 4) Build a universal functional platform, implement a standard normative system, take an efficient central command system as the core, realize the real-time scheduling, aggregation and sharing of various information resources, and drive intelligence to ensure that the big data platform is standardized, orderly and healthy, Safe and sustainable operation.

3.2 Establish a Data Connection

The connection of all things requires a platform. When the big data platform is fully constructed, the real or virtual world can be reconstructed based on big data. Data connection

Table 1. Cross-border data fusion application tools

Scale adaptability	
Supply and demand scale indicators	Market ecological indicators
Land transaction construction area\land launch index	Real estate development investment\urban GDP
Commodity housing area in the current month\Sale area of commodity housing in the month	Real estate development investment\urban fixed asset investment
Pre-sale certificate-area\Market sales area of the month	New building area, floor area ratio, greening rate, number of buildings
Annual total-housing transaction volume\housing start rate	Number of entrepreneurial platforms\city scale growth rate
Staircase ratio, main-area, and-best-selling apartment-types	Gross-Industrial-Product\Domestic and foreign trade deficit
Residential consumption scale of urban residents\other consumption scale	Residential-sales\total-consumption-expenditure
Structural rationality	
Individual-house-purchase-pressure-coefficient\Personal loan interest rate index	Concentration of commercial resources\Favor index of big brands
Stock Index\Development Enterprise Confidence Index	Consumption Diversity Index\Traffic Connection-Index
Price-to-income ratio\financing scale\development cycle	Corporate Brand Communication Power Index\Number of Catering Stores
Existing real estate opening price\similar real estate gelling price\free area	Air Quality Index\Talent Attraction Index
Real estate loans\fixed asset investment price index	Commercial inventory rate\delivery speed\commercial supporting facilities ratio
Capital account income and expenditure\current account income and expenditure	Number of entrepreneurial platforms\city scale growth rate
Temporal dynamics	
Year-on-year growth rate of real estate investment\Year-on-year growth rate of transaction area	Population density Proportion of high-end residential area
Year-on-year price growth\personal income	Residential area planning total-land use-l traffic-distribution
Rental price index\Increase in-intermediary wages	City dynamics\urban population trend
Residential area parking space price, parking space location, parking space supply and demand ratio	Graduate-employment rate\unemployment rate
Construction and decoration price increase\Purchasing Managers Index	Real-time traffic congestion\marriage and divorce rater

is suitable for multi-level data related applications. It is responsible for visual presentation, convenient use, efficient transmission of data information and value conversion

at the front end; multi-level data model construction, underlying data code architecture, technological innovation and data at the back end. The data connection implements the road of data exploration from the table to the inside:

Table level: general media big data with extended data dimensions. For example, new media dissemination of big data, global public opinion big data, network image big data, etc.;

Table two: Big data in vertical industries for mining data value. For example, vertical portals, industry media, UGC in the aggregate industry, etc.;

The first layer: industry private data to ensure data security. For example, industry enterprise OA system, enterprise internal database, enterprise information database, etc.;

The second floor: user portrait data with precise data positioning. For example, user background information data, user usage and purchase behavior data, user scenario data, etc.;

The third layer: Explore the genetic and biological big data of the data interface. For example, gene sequencing and labeling, genetic information decoding, precision medicine and individual services.

The five-layer data association model is from the outside to the inside, and through the construction of a multi-level data closed-loop and generalized, one-stop data platform, from general to vertical, from corporate institutions to individual users, from global picture to gene portrait, continuously extending the data reach. Among them, the surface data is collected by machines and integrated by data application companies through technical means; while the inner data needs to be integrated with relevant government agencies, enterprises, and media. In this way, while strengthening the public data collection and integration infrastructure, it will guide the aggregation of private data in more industries.

3.3 The Realization Path of Big Data Industrialization

Big data industrialization refers to the act of selling big data to the market as a standard product. The core logic of big data operations, the main expectations of big data platforms, the inevitable law of informatization development and other issues all urgently need new information processing methods to follow the evolution route of “data-information-digital-intelligence” and continue to upgrade [8]. Intelligence is an advanced stage of the development of big data industrialization [9]. This stage requires multiple big data companies to build a big data industry ecology and realize the advanced stage of business logic in a cross-border integration way. Cross-border data fusion is an effective catalyst to change the thinking mode of the big data industry in the evolution route, and catalyze the refinement of data. The essence of “General Big Data Platform+” is cross-border data fusion, and “+” is the core. Proposing solutions for cross-border data fusion is the key.

There are numerous and constantly emerging cross-border data fusion application tools. How to select application tools according to specific business needs to achieve efficient and accurate application results is a challenge, especially when many cross-border data fusion application tools have more than a single purpose. We tend to use a single category and choose suitable application tools for different divisions of labor in the big data industry. Table 1 summarizes the existing application tools at home and

abroad, which are roughly divided into system general tools and industry application tools.

4 The Realization of the Big Data Industry Chain

4.1 Core Construction of the Industrial Chain

As a kind of extended industry chain, the big data industry chain refers to linking the related industries of big data products to form a step-by-step production industry, from the most basic to the highest-end core to form a line of production. The essence is different. The industry's associations is between enterprises [10]. Specifically, the big data industry chain refers to the use of modern management models, through standardized data interface design, integrated data collection, data storage, data analysis, data mining, data visualization batch management, combined with various fields to achieve digital and intelligent business model. The interruption or lack of industrial nodes in the industrial chain will affect the industrial supporting capabilities and the failure of industrial transformation and upgrading. Promoting the big data industry chain will inevitably require national policy support, a complete standard system, a healthy market environment, and the concerted collaboration of enterprises to achieve a more optimized allocation of resources within the industry chain for the entire big data industry. Based on the special form of big data, the big data industry chain is set as a “docking mechanism” that is different from the traditional chain-type association relationship, including six dimensions of technology chain, value chain, enterprise chain, supply and demand chain, space-time chain and information chain. Among them, the technology chain is the foundation, the value chain is the goal, the enterprise chain is the guarantee, the supply and demand chain is the key, the space-time chain is the positioning, and the information chain is the adjustment.

Technology chain: including data collection, data storage, data analysis, data mining, data visualization technology iteration and model reconstruction. For example, as the most extensive big data warehouse tool, Hive has simple data structure and high reliability, but it has problems such as poor real-time performance and relatively low efficiency. Continuous technological updates are needed to adapt to massive and diverse information assets with stronger decision-making power, insight and discovery power, and process optimization capabilities.

Value chain: Through understanding users' data needs, a series of related activities are carried out to enhance competitive advantage. The source of these activities is the innovation of knowledge and technology, which can create basic values such as logistics, production, and sales, as well as information push, advertising and other added value to tap hidden customers. The basic principle is that under the premise of meeting the needs of the market, through effective management of the big data industry chain, big data products can realize the basic value while increasing the value as much as possible.

Enterprise chain: An enterprise chain formed by the flow of capital and technology in the life of an enterprise. Enterprises are often not independent but interdependent. Horizontal enterprise linkage is the mutual linkage between enterprises of the same type, and vertical enterprise linkage is the mutual influence between upstream and downstream enterprises.

Supply chain: Including data sources and data requirements, its network model is to obtain data from data sources and process them into data types required by customers.

Space-time chain: reflect the relationship of space-time layout. It is used to reflect the regional centralized information within time t and the regional scattered information within time t .

Information chain: including historical information, real-time information and unknown information. Historical information is used to accumulate the experience required by each link, and real-time information is used to feed back the real-time operation of each link. Both can be used to adjust data collection and storage according to market needs. Unknown information is used to predict the approaching demand of each link in order to maximize the value added of the big data industry chain.

Figure 1 is a three-dimensional and two-dimensional connection diagram of the big data industry chain. The enterprise chain depends on the balance of market supply and demand. Technology adjustment is affected by the development of information under time and space conditions. The enterprise chain, technology chain, supply and demand chain, time and space chain, and information chain work together to achieve maximization of the value chain.

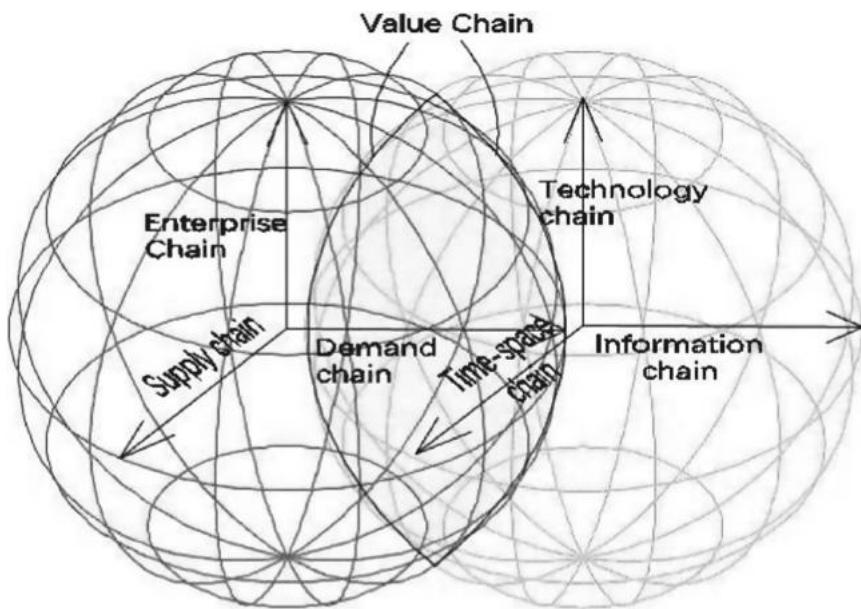


Fig. 1. Three-Dimensional and Two-Dimensional Docking of the Big Data Industry Chain

4.2 Framework and Map of the Industry Chain

In the digital age, the formation goal of the big data industry chain is interconnection, the core connection subject is big data resources, the communication mode between the

connection points is the scene, artificial intelligence, 5G, cloud computing and other technologies are the main forces that quickly form the industry chain. The Internet of Things will become the main battlefield of connection, and ultimately form a main pattern of pluralism and integration.

The structural framework of the big data industry chain is divided into five sections: infrastructure providers, data resource providers, data products and technology providers, application service providers, and support service providers. Infrastructure is a provider of IT infrastructure such as software, hardware, and networks; data resources are government departments, enterprises, institutions, individuals or data circulation platforms that provide data sources; data products and technologies include big data platforms, cloud storage, Data security and related processing analysis, solutions and other software products suppliers; application services are to provide intelligent transportation, smart home, smart medical and other industry applications and other government departments or enterprises; support services include infrastructure, storage analysis, A provider of consulting and maintenance services related to security warnings. Based on the above structural framework, the model of the big data industry chain can usually be described as:

Among them is the big data industry; it is the upstream industry composed of data sources such as media and the Internet; it is the downstream industry guided by big data, such as finance, science and technology, etc.; \times means between the two industries. For example, $A \times B$ means that the development of industry B should take the output of industry A as input, that is, B is the downstream industry of A.

In fact, the big data industry chain is usually due to the differences in the division of labor between companies engaged in the big data industry, and the economic linkages between resources and applications formed by it are not stable [11]. As an economic association with technology as the core, each enterprise, in the process of regulating the industrial chain, the effective solution to alleviate this situation is to use midstream technology as a balance point and set the big data industry chain as the upper, middle and lower reaches. Upstream extension makes the big data industry chain enter the basic industry link, including the acquisition of multiple data sources and data processing algorithms and model research and development links, and downstream expansion will enter the market expansion link, including data reprocessing, data and other industries In combination, the midstream supports the extension and expansion of the upstream and downstream, making the industry chain enter the technology research and development link.

Upstream Basic Layer based on Resources (Data and Infrastructure Owners)

Data and IT infrastructure are respectively the capital and foundation for data resource suppliers to continuously realize monetization, helping some companies to upgrade from order-based to operational-oriented, and keep transactions in a dynamic process of continuously mining data value monetization. Such companies mainly include big data. As the core of the business, companies that reuse big data based on IT infrastructure; companies that use big data as a business tool to improve production efficiency and increase business income or other income based on IT infrastructure; big data that does not have the ability to create data. Intermediaries integrate and utilize data through investigations, crawling, etc. Upstream resources are expected to build a data resource

collaborative management system and big data infrastructure support belt, strengthen the overall management and utilization of industrial big data resources, promote the co-construction and sharing of IT infrastructure, and realize the sharing and integration of industrial big data resources. The entire upstream basic layer includes two frameworks: infrastructure providers and data resource providers.

Midstream Middle-Tier with Technology (Products, Technology Providers) as the Mainstay

The middle reaches of the big data industry chain mainly include data storage, data processing, data analysis and other links, and it is expected to build a big data collaborative processing center. Relevant companies involve software and hardware companies, analysis service companies, information security companies, and solution providers. They determine the database architecture, data organization and management, and the extent and scope of application promotion for big data applications. They usually provide single-point technology and overall, three business models of big data solutions and cloud data solutions. The entire midstream middle tier includes a framework for data product and technology providers.

Downstream Application Layer based on Services (Service Providers)

Compared with resource-based and technology-based companies, application-based companies (application service providers and consulting service providers), especially big data companies in the vertical application field, are more focused on solving industry pain points, forcing software technology, data architecture, and data sharing. The transformation of the method continuously promotes the development and maturity of all links in the big data industry chain, and realizes the realization of the realization of the value of big data. The collaborative application of the big data industry is suitable for building a platform service system for different service objects. It is expected to build big data analysis services such as operation status control, situation prediction, and abnormal mining in various fields, industries, and departments, and industry collaboration and cooperation between services. The aggregation gradually formed. The entire downstream application layer includes two frameworks: application service providers and supporting service providers.

In the digital age, the upper, middle and lower reaches of the big data industry chain play the role of a “navigator” for the industrialization of big data, in order to strengthen the precise implementation of big data industry upgrades. At the same time, it is necessary to run through all links of the industry, including production, circulation, sales, and services, to improve the efficiency of big data resource allocation, and to promote the generation of solutions such as big data industry planning and investment. The big data industry chain map is based on a single enterprise of big data. Based on the structural framework of the industry chain, it analyzes the institutions, products, and technical levels involved in each link of the industry chain, and seeks target suppliers, distributors, and cooperative enterprises, etc., throughout the whole process transmission mechanism [12].

How to have a deeper understanding of the composition and transmission mechanism of the industrial chain? The industry map sorts out the industry chain information through the overall industry information, and qualitatively characterizes the relevance of various

industries according to the logic of the upstream basic layer, midstream middle layer, and downstream application layer, grasps the linkage of the industry, and realizes the coordinated promotion of the development of each link of the big data industry chain. The multi-dimensional, multi-level and three-dimensional big data industry map is a model example similar to the snowflake model. It consists of a fact table (Fact, presenting summary information of the industry map) and a series of dimension tables (Dimension, presenting detailed information of the industry map). The fact table is connected to the primary keys of all dimension tables through foreign keys, as shown in Fig. 2.

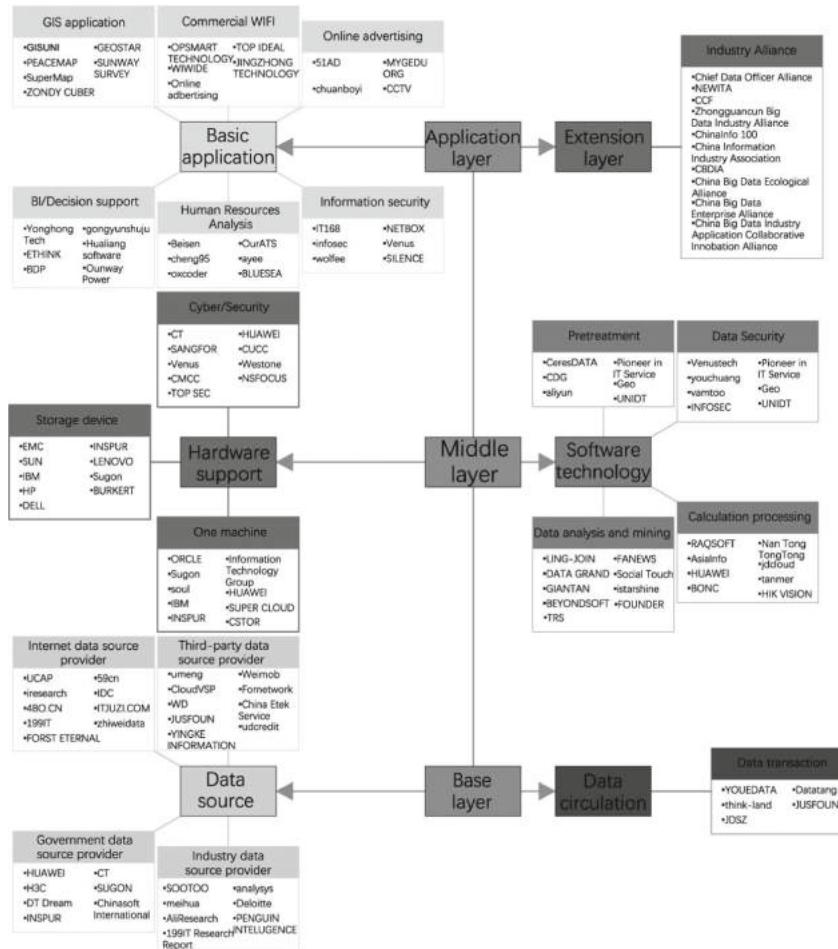


Fig. 2. Multi-Dimensional, Multi-Level and Three-Dimensional Big Data Industry Map

4.3 The Transformation and Upgrading Path of the Industrial Chain

The society at home and abroad is in the midst of a technological revolution marked by technologies such as big data, cloud computing, virtual reality, blockchain, and 3D printing. This revolution takes big data as the core, connects the information technology that promotes the advancement of society, and makes the social development move towards integration, intelligence, innovation, and integration [13]. Therefore, the realization of the big data industry chain is inseparable from the development of information technology.

Emerging technologies such as 5G, Internet of Things, cloud computing, and artificial intelligence are rapidly moving towards large-scale commercial use, and play different roles at different levels of the big data industry chain. In terms of upstream resources, they mainly rely on 5G, Internet of Things and other technologies to obtain; in the midstream technology. It mainly relies on artificial intelligence, cloud computing, blockchain technology, etc.; in downstream applications, it mainly applies artificial intelligence, the Internet of things, and augmented reality.

“Artificial Intelligence+” Technology Model

Artificial intelligence is the core driving force for the transformation and upgrading of the big data industry chain. Its development has gone through the stage of computational intelligence, perceptual intelligence and cognitive intelligence [14]. The key technology is mainly manifested in the representation method from computing to cognition to perception. From the perspective of the industrial chain, the “artificial intelligence+” model uses big data as the source of power, and the innovative activities of artificial intelligence as the framework. “Artificial intelligence + horizontal” realizes self-adaptation, self-learning, self-perception, self-determination, and self-determination. Feedback, self-adjusting technology model, “artificial intelligence + vertical” to realize the autonomous optimization technology model of deep learning algorithms. The “artificial intelligence+” model makes distributed management, personalized experience, and natural human-computer interaction tend to be autonomous, social, and logical, triggering a deep transformation of the big data industry chain: the industrialization of big data as data capital replaces traditional materials Investment; the intelligent services of each structural framework in the big data industry chain replace traditional manual services; in particular, the business computing and data caching capabilities in the upstream, mid-stream and downstream of the big data industry chain are closer to the user terminal Instead of the separation of traditional networks and services, this is a new opportunity for edge artificial intelligence to upgrade the big data industry chain.

Relevant information shows that the global Internet of Things device data has approached 20 billion units. Such application scenarios increasingly need to make real-time adjustments to the development of the big data industry chain. New processing requirements for big data will inevitably arise during the adjustment process. In fact, it is impossible to always transmit all data to the cloud for artificial intelligence. Processing, which gave birth to a new form of edge intelligence. Edge devices can access software packages related to the selected model from the shared resource library without having to rely too much on the cloud. At the same time, its inherent strong security, preset inference

functions, and data transfer restrictions between networks make The midstream technology and downstream services of the big data industry chain have flexibly constructed customized solutions, which have the advantages of network delay advantages, network distribution advantages, security and reliability improvements, and flexible industrial layout, realizing network access channels. The key leap from transformation to informatization enabling platform [15].

Edge artificial intelligence provides conditions for the deployment of the smart big data industry chain: First, it takes upstream resources as the entry point to extend the layout to midstream technology and downstream applications, that is, relying on big data resources in the data resource collaborative management system and big data infrastructure support belt Increasing R&D investment in big data collaborative processing centers and big data solutions to build application scenarios in the horizontal and vertical fields of the industry; the second is to use midstream technology as the entry point to expand upstream and downstream business areas, That is, relying on the application and development of natural language processing, machine learning, deep learning, emotional analysis, etc. to increase research on computing power and algorithms, so that intelligent evaluation, human-computer interaction, and semantic mining reach the height of a new round of technological revolution, Expand the application scenarios of industry segments and vertical fields; third, take downstream applications as the entry point, and increase business layout upstream of the industry chain, that is, relying on the collaborative application of the big data industry to increase the predictive and intelligent Investment in the provision of decision-making, the perception of gathering knowledge, etc. to enhance the application scenarios of the industry's horizontal and vertical fields.

“5G + Internet of Things” Technical Model

5G and the Internet of Things play a key role in promoting and expanding the development of data and computing power respectively. First, at the data level, 5G promotes the generation of massive amounts of big data, and the Internet of Things connects these data better. With the support of 5G networks, the connection between things has become a social development trend, and the number and quality of big data connections have been improved by orders of magnitude. Secondly, at the level of computing power, 5G further liberates the real-time analysis and processing capabilities of big data and a large number of cloud application capabilities. The Internet of Things is based on an extended and expanded network, which exchanges and communicates calculated data in real time [16]. Therefore, 5G promotes the application of scenarios in the big data industry chain, and the Internet of Things promotes the large-scale generation and construction of scenarios in the big data industry chain.

The “5G + Internet of Things” technology model provides conditions for the broader deployment of the smart big data industry chain: one is based on hardware, and the realization of a digital data acquisition perception layer in the upstream resource layer; the other is based on both hardware and software. The data dissemination network layer that realizes data interconnection in the midstream technology layer; the third is

based on software, and the intelligent data processing application layer is realized in the downstream application layer.

“Cloud Computing + Blockchain” Technology Model

The big data industry chain needs the dual support of super computing power and the flexibility of information resource management, processing, and application. Cloud computing, as a type of distributed computing, has spatio-temporal elasticity. Thousands or even tens of thousands of machines are placed in a “pool” through “pooling” and “cloudification”. How much CPU, memory, and hard disk do users need? The “virtual computer” only needs to be scheduled to find and use the required information resources in the “pool”. At the same time, the ultra-large-scale concurrent data processing demand caused by the exchange of massive node data poses a huge challenge to the existing data center’s computing, storage, and stable service capabilities. Once maliciously attacked, the private data stored in the central server will be leaked. The blockchain technology with centralized computing characteristics naturally provides a type of solution for high concurrency problems. Its decentralized distributed database can solve the security, privacy and other technical problems faced in the Internet of Things network [17].

The “cloud computing + block chain” technical model is used to solve the distributed problem with task decomposition as the core in the big data processing process, the transparent problem with the alliance chain + private chain as the core, and the workflow reconstruction as the core Parallel problems, collaborative problems with algorithm scheduling as the core, etc.: add a layer of “application elasticity” on top of IaaS, namely PaaS and SaaS, to meet the “application elasticity” needs of the big data industry chain; add on top of IaaS A layer of “application mining”, namely, DaaS and KaaS, to meet the “application mining” needs of the big data industry chain; add a layer of “application flexibility + mining” on top of blockchain as a service (BaaS) to reduce big data The deployment cost of the industrial chain and the improvement of the collaboration efficiency of the big data industrial chain.

5 Conclusions and Prospects

This article builds a new reference architecture from the industrialization of big data to the big data industry chain to promote the comprehensive and high-speed construction of the big data industry. Today, big data is still affected by data acquisition at the data resource layer, data processing at the data technology layer, high variability in the data service process at the data application layer, difficulty in quantification, and lack of uniform standards. How to structure the industry pivot? Accurately predicting user needs and how to use information technology to promote industrial development and other related issues may become major obstacles to driving the upgrade of the big data industry. However, through strengthened cooperation, big data can only be used as a tool to meet the functional needs of various lines and industries, and work together to build a common big data platform to achieve maximum efficiency and sustainable application of big data, achieving the ultimate goal of mutual benefit and win-win.

At the same time, this also brings unprecedented opportunities and challenges. Future work will be fully carried out from the following two aspects:

- 1) Data problem. There are many sources of big data, and most of them come from heterogeneous environments. The completeness, consistency, and accuracy of the data source obtained in this case cannot be guaranteed. In further work, it is necessary to more steadily integrate big data in horizontal and vertical fields, and realize data refinement, integration, serviceability and arrival rate. In addition, it is necessary to develop more data platform tools for data manipulation, including efficient and intelligent data processing technology; segmented, unbounded, and virtualized data application scenarios; automated, diversified, and accurate data models; dynamic, personalized, multidimensional and other data presentations have made data platform tools tend to be vertical, scene-oriented, efficient, real-time, cloud-based, integrated, and open.
- 2) Analyze the problem. The development of emerging technologies is becoming a main thread running through the transformation and upgrading of the big data industry. The process of transformation and upgrading of the big data industry is an interactive process of industrial resources, industrial applications and industrial technologies. However, because the acquisition of resources is extensive, the research and development of technology is professional, and the needs of users are dynamic and variable, future work needs to study and verify the multi-level and multi-granularity tasks in the interaction process of industrial resources, industrial applications and industrial technology. Distribute and implement a user-centric system framework, and grasp social needs in real time, such as hot spot forecasting, trend judgment, etc.

References

1. Li, Y., Zhang, S.: The path to the transformation and upgrading of traditional industries driven by big data—Based on the perspective of big data value chain. *Sci. Technol. Manag. Res.* (07), 156–162 (2019)
2. Li, H.: The theoretical mechanism, practical basis and policy choice of big data to promote the high-quality development of my country's economy. *Economist*, (03), 52–59 (2019)
3. Wang, T.-M., Tao, Y., Liu, H.: Current researches and future development trend of intelligent robot: a review. *Int. J. Autom. Comput.* **15**(5), 525–546 (2018). <https://doi.org/10.1007/s11633-018-1115-1>
4. Hu, X., Meng, Z., Wang, X., Lin, Q.: A review of foreign scientific data analysis platforms. *Res. Inf. Technol. Appl.* **10**(04), 56–62 (2019)
5. Deng, Z., Chen, L., He, T., Meng, T.: A regional big data industry development strategy formation method and case study. *Sci. Technol. Manag. Res.* (12), 160–170 (2017)
6. Yong, S., et al.: Research on construction of new intelligent municipal facilities and promoting strategy of industrialization. *Urban Roads Bridges & Flood Control* (2018)
7. Wang, J., Wu, F.: The driving force of the transformation and upgrading of the geographic information industry. *J. Wuhan Univ. (Inf. Sci. Ed.)* **44**(01), 10–16 (2019)
8. Peng, Y., et al.: Design and modeling of survivable network planning for software-defined data center networks in smart city. *Int. J. Commun. Syst.* **31**, e3509 (2018)
9. Xie, W., Fan, B., Dong, C.: Comparative analysis of the development of big data industry at home and abroad. *Mod. Inf.* **39**(09), 113–121 (2018)
10. Zhou, Y., Liu, Y.: Research on the influencing factors of big data industry development. *Mod. Inf.* **37**(08), 129–134 (2017)

11. Wang, Q., Sui, Q., Yao, P.: Evolutional relationship between industrialization and urbanization under environmental constraints in China: DEA analysis perspective. *Scientia Geographica Sinica* (2017)
12. Youjia Industrial Big Data. Industrial Big Data Center, Industrial Big Data-“Industry Atlas” [DB/OL]. 17 May 2019
13. Zou, D.: Philosophical thinking on artificial intelligence and its modernity dilemma. *J. Chongqing Univ. (Soc. Sci. Ed.)* **11**(30), 1–12 (2019)
14. Zhang, H., Wang, J.: Application of big data technology in artificial intelligence. *Electron. Technol. Softw. Eng.* (21), 242–243 (2019)
15. Goldes, S., et al.: Building a viable information security management system. In: 2017 3rd IEEE International Conference on Cybernetics (CYBCONF). IEEE (2017)
16. Liu, W., Yuan, J.: Big data thinking approach to artificial intelligence problems. *J. Xinjiang Norm. Univ. (Philos. Soc. Sci. Ed.)* **39**(02), 120–125 (2018)
17. Liu, H.: “Big Data + Blockchain” sharing economy development research—Based on the theory of industrial convergence. *Technoecon. Manag. Res.* (01), 91–95 (2018)



Using Correlation and Network Analysis for Researching Intellectual Competence

Sipovskaya Yana Ivanovna^(✉)

Institute of Psychology of the Russian Academy of Sciences, Yaroslavskaya Street, 13,
129366 Moscow, Russia

Abstract. The article examines nonparametric models of the productivity of the intellectual activity of older adolescents. The study involved 90 older students (15 years old). Methodological base of the research: “Conceptual synthesis,” “Moods,” “Methods for diagnosing the level of reflexivity,” “Comparison of similar drawings”, “Interpretation.” The study results allow us to conclude that network modeling showed the best results among all the methods of nonparametric statistical analysis. The structure of the network model of the construct of intellectual competence showed that the core of this construct consists of abilities characterized by the function of generating a new context (conceptual abilities), an implicit intellectual suspicion of the direction of finding a solution in a situation of uncertainty (intentional abilities, mentality), arbitrary regulation of their intellectual activity (reflexivity). In contrast, cognitively more straightforward abilities (sentences of factual and interrogative type) and involuntary metacognitive abilities (indicators of reflectivity) are on the periphery. However, some of the latter, namely, indicators of reflexivity in the form of a cognitive tempo, are nevertheless associated with sentences of emotional-evaluative personality and content types.

Keywords: Data analysis · Parametric · Linear methods

1 Introduction

Intellectual competence is a particular form of organizing individual mental experience [14]. This understanding of intellectual competence is consistent with the results of various psychological studies. In particular, the achievement of competence presupposes the formation of systems of practical and mental skills that a person masters in the course of a long, “conscious practice” [9]. Experts acquire intellectual competence only after spending a lot of time in favor of studying the relevant subject area, accumulating decision-making experience, etc. The hallmarks of “deliberate practice” are a high level of motivation for learning (cognitive need), constant feedback on the assessment of the correctness of their actions or their erroneousness (criticality, reflexivity); depth and thoroughness of material processing; initiative and independence. In recent years, the critical importance of motivation and value orientations in intellectual competence development has been demonstrated [20].

Thus, the components of intellectual competence are:

1. conceptual abilities, which provide a particular type of organization of subject knowledge about the subject area and ways of transforming this knowledge in solving the assigned tasks [14, 15] etc.;
2. metacognitive abilities responsible for intellectual self-regulation [15, 19];
3. intentional abilities manifested in individually-specific cognitive preferences and underlying the accumulation of implicit knowledge, on which, in turn, explicit, explicated knowledge is formed [21].

Thus, the research analysis in the psychology of competence made it possible to formulate a hypothesis that the critical components of competence are conceptual, metacognitive, and intentional abilities. Certain thinking qualities are also associated with intellectual competence (cognitive need, the selectivity of interests, rationality, criticality, creativity, reflexivity, flexibility, independence, dialogism, general mental culture). Accordingly, an urgent scientific task is to reveal the structure of intellectual competence, show the relationship of its constituent components, and how important it is to measure their participation in effective intellectual activity. It is for a complete definition of the construct “intellectual competence,” during which conceptual (conceptual), metacognitive and intentional abilities as components of intellectual competence were studied in more detail (Fig. 1):

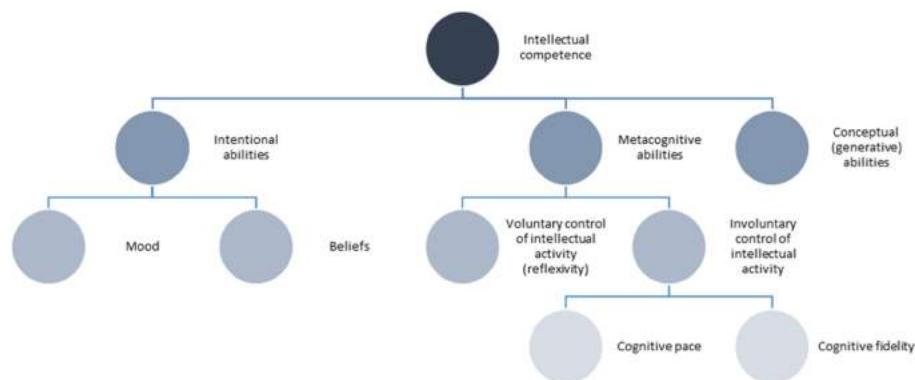


Fig. 1. The Structure of Intellectual Competence

This approach certainly does not exclude the role of other components. However, this model basis on the idea of a necessary condition - the presence of subject knowledge, individual conceptual experience. Simultaneously, the ability to reflect conceptually and operate with generalized concepts is the highest stage of intellectual development.

Concepts are integral cognitive structures characterized by different ways of coding information, the hierarchical nature of the organization of semantic features, and the complexity of the cognitive composition, including sensory-emotional impressions [14]. The participation of sensory-emotional impressions in the composition of intellectual

competence is a prerequisite for tacit knowledge, which, in our opinion, is an indicator of intentional abilities. Also, following the studies of J. Raven [20], Ericsson et al. [9] and other authors, we have identified the qualities of thinking that are related to intellectual competence: cognitive need, the selectivity of interests, rationality, criticality, creativity, reflexivity, flexibility, independence, dialogism, general mental culture.

Thus, there is reason to assume a particular complex of related mental formations that characterize intellectual competence, including intellectual competence and its variety - school competence (the process of school education among students forms this type of competence). We analyzed the “conceptual, metacognitive, and intentional structure” of intellectual competence, which implies considering the outstanding abilities as components of the construct of intellectual competence. This research is devoted to constructing a conceptual model of intellectual competence (productive intellectual activity) in terms of one’s semantic experience.

1.1 Research Questions

Hypotheses:

1. to reveal the specifics of the construct of intellectual competence, determined in the multidimensional context of test changes, the following methods of statistical analysis are necessary and sufficient: correlation analysis (Spearman’s method) and network modeling;
2. there is a connection between the level of development of conceptual abilities (in terms of conceptual abilities measured by the methodology “Conceptual synthesis”), metacognitive abilities (in terms of “reflexivity,” measured using the “Methodology for diagnosing the degree of development of reflexivity” for arbitrary metacognitive abilities and in terms of cognitive style “impulsivity/reflectivity,” as measured by the method “Comparison of similar patterns” for involuntary metacognitive abilities) and intentional abilities (the ability to mentality and beliefs, measured by the method “Mood”) and indicators of the formation of intellectual competence (measured in terms of narrative - interpretation moral dilemma) among 9th-grade students of a comprehensive school;
3. the operationalization of intellectual competence in terms of indicators of the complexity of generated texts (texts of interpretation of a moral dilemma) is productive. It allows one to describe intellectual competence as a meta-ability, in which the subject’s different-level mental resources are presented (intellectual abilities of various types).

1.2 Purpose of the Study

The subject of this study is the indicators and composition of the construct of intellectual competence. The study’s object is students of the 9th grade of secondary schools.

Purpose of the study: disclosing the specifics of intellectual competence in older adolescence using a statistical analysis of research indicators.

2 Study Participants

Students of ninth grades took part in the study, totaling 90 older teenagers (Fig. 2 and 3).

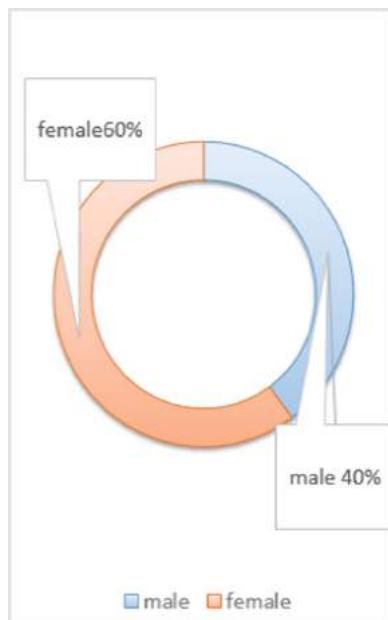


Fig. 2. Study participants. Sex differentiation

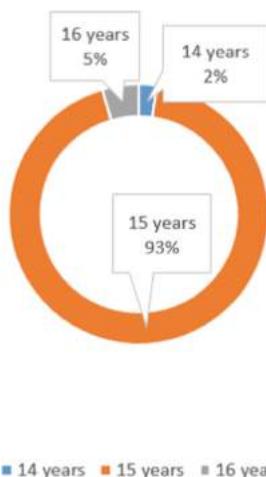


Fig. 3. Study participants. Age differentiation

3 Research Methods

3.1 Methodology for Diagnosing Conceptual Abilities. “Conceptual Synthesis” [14]

Indicator: the level of formation of conceptual abilities (the success of constructing absent connections between concepts), Concept.

3.2 Methods for Identifying Metacognitive Abilities

3.2.1 “Methods for Diagnosing the Level of Reflexivity” [13]

Indicator: the level of reflexivity formation as an aspect of metacognitive (voluntary) abilities, Voluntary.

3.2.2 “Comparison of Similar Drawings” by J. Kagan [12]

Indicators of impulsivity/reflexivity as an aspect of metacognitive (involuntary) abilities: 1) latent time of the first response (sum), InvoluntaryTime; 2) a total number of errors, InvoluntaryMistakes.

3.3 The Author’s Method of Diagnostics of Intentional Abilities “Mood”

Indicators: 1) mentality, Mindset; 2) beliefs, Confidence.

3.4 Author’s Methodology for Assessing Intellectual Competence “Interpretation”

Along with an overall assessment of the text’s complexity, we held a more thorough analysis of the texts. The units of the analysis were sentences as text units.

Indicators of the methodology “Interpretation”: 1) the score received by each participant in the study, interpretation; 2) the number of sentences of different types, a-g (Ability to generate sentences of factual, systematizing, arguing, interrogative, interpreting, emotionally evaluative content, emotionally evaluative personality type).

We use correlation analysis and network modeling using the R scripting language to process the results.

3.5 Research Techniques

We carried out correlation analysis and network modeling to establish the structure of the construct of intellectual competence. Weighted network correlation analysis, also known as weighted gene co-expression network analysis (WGCNA) or network analysis, can be used when it is necessary to reduce the dimension of variables without losing significant ties in data mining. He defines constructs as a system of cause-and-effect relationships between a network’s components, where it is not so much the causal connection that matters as the formed network architecture [2, 5, 6].

Furthermore, network analysis examines the structure and dynamics of this or that construct network. All research variables are not measures of a construct, but their part. That is a component of a system whose properties are not the aggregate components, its components - a manifestation of emergence [3, 4, 7, 10, 16, 22]. Network modeling of this type is based on the calculation of partial correlations between variables and held on most multivariate datasets. The method allows one to determine modules (clusters), inter-module hubs, and network nodes concerning belonging to a module, study the relationship between co-expression modules, and compare the topologies of various networks (differential network analysis). Specialized regularization methods construct pairs of partial correlations. Regularized (weighted) private correlation networks have the following advantages: • building a network (based on a soft threshold of the correlation coefficient) preserves the continuous nature of the original information about the correlation, while the dichotomous division of information and (hard) choice of the threshold can lead to information loss; • weighted correlation networks facilitate geometric interpretation based on angular interpretation of correlations; • The obtained network statistics improve the results of standard methods of in-depth data analysis, such as cluster analysis; • correlation networks allow the use of economical parameterization (in terms of modules and module membership). Graph theory is the base of network modeling methods. Network models are precise and consider the object of research as a single complex of interrelated components. The used network model of the logical-mathematical description makes it possible to algorithmize the system parameters' calculations. Network simulation of random measurements is an essential method for the analysis of multivariate random measurements. The network itself is a graph composed of nodes connected by edges. Accordingly, the variables of the network turn out to be connected in the form of interdependent paths of transitions (edges) to each other (nodes) [1]. In general, there are two types of links that can be present in the network and represent as graph edges: - the edge can be directed, in which case it has an arrow at one end indicating a one-way effect, - or the link can be undirected, which indicates some mutual relationship. Modeling undirected networks use pairwise Markov random fields. [8]. A node represents each variable, and many edges connect the nodes in these models. Links between nodes - the “weights” of the edges of the network and the edge of an edge are always a nonzero number because zero weight indicates no edge. The sign of the edge weight (positive or negative) indicates the type of interaction, and the absolute value of the edge weight indicates the strength of the bond effect between nodes. The reciprocal of the edge's weight defines the length of an edge in a network. The distance between two nodes is equal to the sum of all edges' lengths on the shortest path between two nodes [17]. One can assess individual nodes' importance in a network by examining nodes' centrality using the specialized Fruchterman Reingold algorithm [11]. Network visualization is an abstract representation of high-dimensional space in two dimensions. But 2D visualization cannot correctly reflect the model's actual space since the metric distance between the placement of nodes in 2D space has no direct interpretation, for example, with multidimensional scaling. Graph theory has developed several methods to more objectively quantify, which node is the most central in a network [18]:

- The node strength, also called the degree in unweighted networks [17], simply adds the strength (weight) of all associated edges to the node; - if the network consists

of partial correlation coefficients, the strength of the node is equal to the sum of the absolute partial correlation coefficients between this node and all other nodes.

- Closeness takes the reciprocal of all shortest paths between one node and all other nodes in the network.
- Strength of a node indicates how strongly a node is directly connected to other nodes in the network - proximity - how strongly a node is indirectly connected to other nodes.
- Betweenness tells how many of the shortest paths between two nodes go through the node in question; - the higher the intermediate, the more critical is the node connecting other nodes.

4 Results

4.1 Correlation Analysis

Figure 4 shows the general correlation matrix of the studied variables in the form of a correlogram. Figure 5 presents the general table of correlations indicating the distribution of intellectual competence indicators, conceptual and intentional abilities.

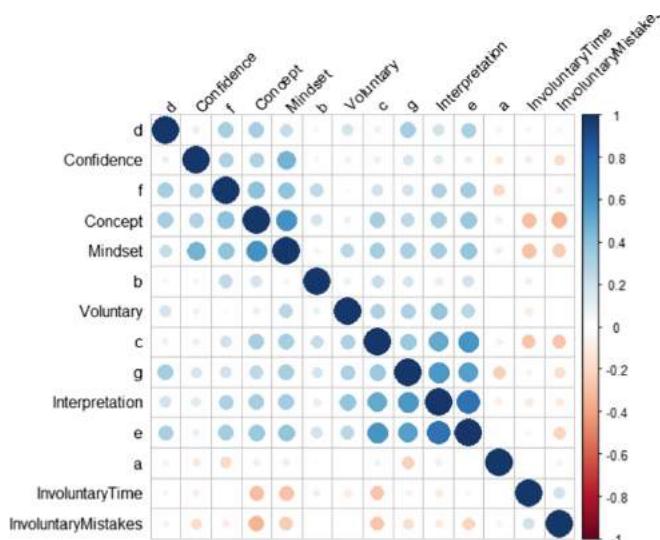


Fig. 4. Correlogram of indicators of intellectual competence, conceptual, metacognitive, and intentional abilities

According to the correlation analysis results, which included indicators of intellectual competence and conceptual abilities, represented in this study by conceptual abilities, a significant correlation was between the general indicator of intellectual competence and conceptual abilities.

According to the results of the correlation analysis of indicators of intellectual competence and metacognitive abilities, there is reason to assume the significance of the

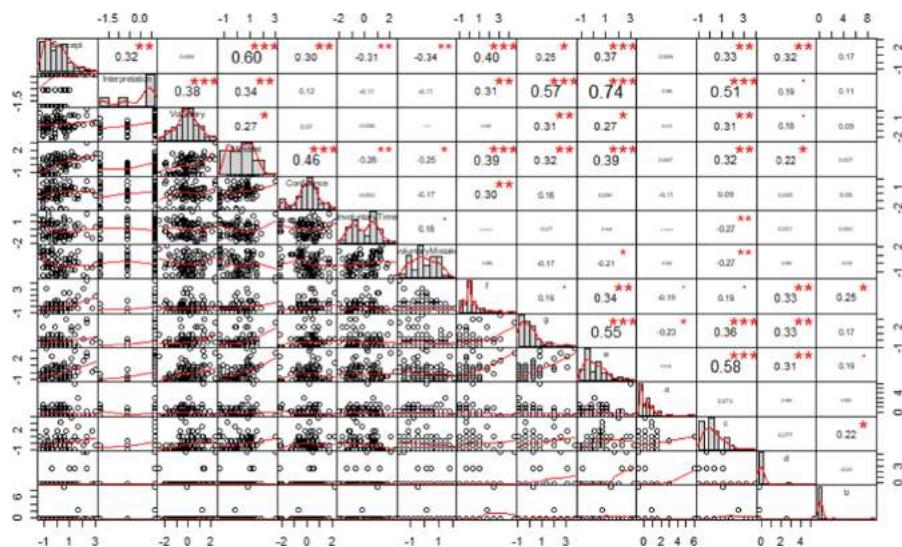


Fig. 5. General correlations indicating the distributions of indicators of intellectual competence, conceptual and intentional abilities

contribution of both voluntary and involuntary metacognitive abilities to the structure of intellectual competence in this age period. Nevertheless, we have revealed the complexity of the metacognitive composition of the regulation of intellectual activity in older adolescence, the multilevel principle of the formation of metacognitive abilities in a given age period, or the uneven participation in intellectual activity productivity of voluntary and verbalized intelligent control and involuntary action. Thus, voluntary and involuntary intellectual control belonging to the metacognitive level of intellectual activity in the construct of intellectual competence was. However, the revealed correlations differ, and this difference depends on the type of manifestation of intellectual competence, the very construct of which is heterogeneous.

Considering the results of a correlation analysis of indicators of intellectual competence and intentional abilities (mentality and beliefs), the importance of their contribution to the productivity of intellectual activity was argued. Simultaneously, there was a recorded heterogeneity of such a communication structure, which, presumably, can be explained by differences in the functional load of each of these abilities. Besides, attention is drawn to the sentences themselves' various cognitive and emotional complexity: from a cognitively and emotionally simple sentence of a factual type to a cognitively complex sentence of an arguing type or an emotionally capacious sentence of an emotionally evaluative personality type. In this area, mental experience resorts to the selective selection of resources to implement this or that mental behavior.

4.2 Network Analysis

Figure 6 and 7 present the results of studying the network structure of the construct of intellectual competence in the context of conceptual (conceptual), metacognitive and intentional abilities, and categorical abilities through network modeling.

Thus, there is reason to conclude about the heterogeneity of intellectual competence's construct due to the functional load and cognitive complexity of the components that make up this Concept's system.

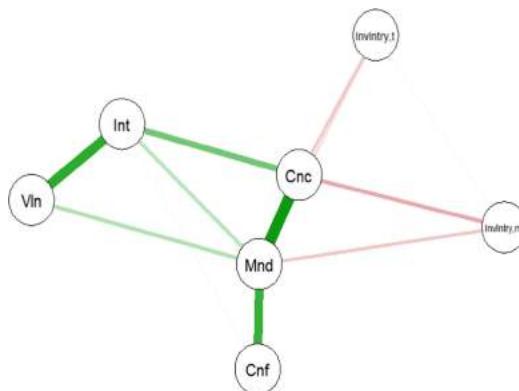


Fig. 6. Network Model of the Construct of Intellectual Competence (General Indicator)

Notes:

- Int - manifestations of intellectual competence
- Vln - reflexivity
- Mnd - mentality
- Cnf - persuasion
- Cnc - conceptual abilities
- Invlntry.t - involuntary metacognitive abilities, cognitive pace
- Invlntry.m - involuntary metacognitive abilities, cognitive accuracy

The simulation results allow us to conclude that the core of intellectual competence consists of abilities characterized by the function of generating a new context, an implicit intellectual suspicion of the direction of finding a solution in a situation of uncertainty, arbitrary regulation of one's intellectual activity. In contrast, cognitively more straightforward abilities (sentences of factual and interrogative type) and involuntary metacognitive abilities are on the periphery. The network structure of the construct of intellectual competence indicates the importance of metacognitive (reflexivity) and conceptual (conceptual) abilities, as well as intentional abilities in terms of moods and beliefs (when analyzing individual sentences). We established the role of involuntary metacognitive abilities (cognitive tempo) only with a more detailed analysis of interpretation texts (connection with sentences of emotional and evaluative personality and

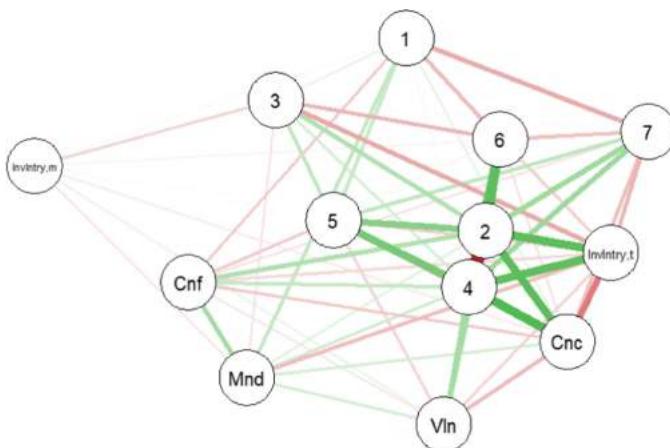


Fig. 7. A network model of the construct of intellectual competence (private indicators of intellectual competence in terms of the ability to generate sentences of various types)

content types). Thus, according to the results obtained, there is reason to conclude that by the age of adolescence, conceptual thinking has been formed sufficiently to ensure a qualitative increase in a teenager's intellectual resources. By this age stage, it becomes possible to turn the mental space most fully, despite the established connections and relationships' heterogeneity.

5 Findings

The data obtained in this study allow us to draw a number of general conclusions.

1. Among all the methods of statistical analysis used in the study, among which were used: correlation analysis (Spearman's method) and network modeling, the network analysis method showed the best results.
2. It is shown the importance of using, along with the classical method of correlation analysis, also network analysis, which allows you to build a model of the investigated construct in its most significant components and eliminate insignificant components from the analysis that can "noise" the results.
3. Conducted using statistical methods, the analysis made it possible to obtain the following significant psychological results regarding the relationship between the indicator of intellectual competence, measured in terms of the characteristics of the mental narrative, with indicators of conceptual abilities, voluntary and involuntary metacognitive abilities in terms of the level of formation of reflexivity and reflectivity, respectively, intentional abilities as the ability to rely on intuitive attitudes in looking for an answer in a situation of uncertainty.

Thus, intellectual competence is a factor in the integration of intellectual abilities of different levels of mental activity organization at the stage of senior adolescence.

1. The structure of the network model of the construct of intellectual competence showed that abilities characterized by the function of generating a new context (conceptual abilities), an implicit intellectual presentiment of the direction of finding a solution in a situation of uncertainty (intentional abilities, mentality), arbitrary regulation of their intellectual activity (reflexivity) present the core of this construct. In contrast, cognitively more straightforward abilities (sentences of factual and interrogative type) and involuntary metacognitive abilities (indicators of reflexivity) are on the periphery. However, some of the latter, namely, indicators of reflectivity in the form of cognitive tempo, are nevertheless associated with sentences of emotionally evaluative personality and meaningful types.
2. Operationalization of intellectual competence in terms of indicators of the complexity of generated texts (texts of interpretation of a moral dilemma) made it possible to describe intellectual competence as meta-ability. The structure of intellectual competence includes conceptual, metacognitive, and intentional abilities. The text is such a mental product, which presents the different-level mental resources of the subject (intellectual abilities of different types).

6 Conclusion

This paper considers the most promising approach to the study and understanding the structure of intellectual competence - a particular form of organization of individual mental experience. This study describes the construct of intellectual competence using the example of the intellectual (school) competence of older adolescents in educational activities (the success of writing the interpretation of a moral dilemma in terms of the complexity of the generated author's text). We identify conceptual, metacognitive, and intentional abilities as components of intellectual competence.

Conceptual abilities as manifestations of conceptual experience and metacognitive abilities in terms of reflexivity and reflectivity and intentional experience, namely, mindsets and beliefs, have various contributions to intellectual competence.

According to the analysis, empirical data found support for all hypotheses.

7 Future Work

Nevertheless, some issues in the framework of this work were not touched upon or covered only superficially. They deserve a separate careful discussion. Such promising areas include, for example, the study of the structure of conceptual experience for its hierarchical organization and the study of the impact on intellectual competence of such an essential factor as motivation. Along with these trends, it is essential to study the manifestation of individual intellectual self-regulation of "experts" as another essential component in the structure of intellectual competence. Discussion of the specifics of methods and methodological techniques for studying the construct of competence and its components is also relevant.

Prospective studies focused on solving these issues will further disclose the psychological mechanisms of competence, improving the tools for its measurement. In general, competence is becoming an essential task in psychological science since the manifestations of people's intellectual competence lead to society's development.

Acknowledgments. State assignment for the IP RAS No. 0159-2019-0008 (Institute of Psychology of the Russian Academy of Sciences).

References

1. Artemenkov, S.L.: Initiative-semantic model of divergent creativity. *Psychol. Sci. Educ. psyedu.ru*. **3**, 1–15 (2012). http://psyjournals.ru/psyedu_ru/2012/n3/55540.shtml. Accessed 30 Dec 2020
2. Artemenkov, S.L.: Network modeling of psychological constructs. *Model. Data Anal.* **1**, 9–28 (2017)
3. Barrett, L.F.: Bridging token identity theory and supervenience theory through psychological construction. *Psychol. Inq.* **22**, 115–127 (2011)
4. Barrett, L.F.: The future of psychology: connecting mind to brain. *Perspect. Psychol. Sci.* **4**, 326–339 (2009)
5. Borsboom, D.: A network theory of mental disorders. *World Psychiatry* **16**, 5–13 (2017)
6. Dalege, J., Borsboom, D., Van Harreveld, F., Van Den Berg, H., Conner, M., Van Der Maas, H.L.J.: Toward a formalized account of attitudes: the causal attitude network (CAN) model. *Psychol. Rev.* **123**, 2–22 (2016)
7. De Schryver, M., Vindevogel, S., Rasmussen, A.E., Cramer, A.O.J.: Unpacking constructs: a network approach for studying war exposure, daily stressors and post-traumatic stress disorder. *Front. Psychol.* **6**(1896), 1–10 (2015)
8. Epskamp, S., Rhemtulla, M.T., Borsboom B.: Generalized network psychometrics: combining network and latent variable models (2017). <https://arxiv.org/pdf/1605.09288.pdf>. Accessed 30 Dec 2020
9. Anders Ericsson, K., Krampe, R., Tesch-Römer, C.: The role of deliberate practice in the acquisition of expert performance. *Psychol. Rev.* **100**(3), 363–406 (1993)
10. Fingelkurts, A.A., Fingelkurts, A.A.: Making complexity simpler: multivariability and metastability in the brain. *Int. J. Neurosci.* **114**(7), 843–862 (2004)
11. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Softw.: Pract. Exp.* **21**(11), 1129–1164 (1991)
12. Kagan, J., Rosman, B.L., Day, D.A.J., Phillips, W.: Information processing in the child: significance of analytic and reflective attitudes. *Psychol. Monogr.* **78**(1), 1–37 (1964)
13. Karpov, A.V.: Reflexivity as a mental property and the method of its diagnosis. *Psychol. J.* **24**(5), 45–57 (2003)
14. Kholodnaya, M.A.: The psychology of intelligence: paradoxes of research. Peter, SPb (2002)
15. Kholodnaya, M.A.: The psychology of conceptual thinking: from conceptual structures to conceptual abilities. Institute of Psychology RAS (2012)
16. McNally, R.J., Robinaugh, D.J., Wu, G.W.Y., Wang, L., Deserno, M.K., Borsboom, D.: Mental disorders as causal systems: a network approach to posttraumatic stress disorder. *Clin. Psychol. Sci.* **3**, 836–849 (2015)
17. Newman, M.E.J.: Networks. An Introduction. Oxford University Press, Oxford (2010)
18. Opsahl, T., Agneessens, F., Skvoretz, J.: Node centrality in weighted networks: generalized degree and shortest paths. *Soc. Netw.* **32**, 245–251 (2010)
19. Sternberg, R. (ed.): Practical intelligence. Peter, SPb (2002)
20. Raven, J.: Competence in Modern Society: Identification, Development and Implementation (translated from English). Kogito-Center (2002)
21. Sultanova, L.B.: The problem of implicit knowledge in science. USPTU Publishing House, Ufa (2004)
22. Van Orden, G.C., Kloos, H., Wallot, S.: Living in the pink: intentionality, wellbeing, and complexity. In: Hooker, C. (ed.) *Philosophy of Complex Systems*, North-Holland, Amsterdam (2011)



A Disease Similarity Technique Using Biological Process Functional Annotations

Luis David Licea Torres^(✉) and Hisham Al-Mubaid

University of Houston-Clear Lake, Houston, TX, USA
{liceatorres,hisham}@uhcl.edu

Abstract. Disease similarity methods are important for understanding disease mechanisms and relationships. Disease similarity results can be utilized in medical applications like drug repurposing and therapy development. In this paper, we present a new method for measuring disease similarity using biological process functional annotation. We measure the similarity between diseases by employing a weight function on the shared biological processes of disease genes. We evaluated the method in several experimental settings, and the results are encouraging. We found a strong and positive correlation between the weight of shared biological process functions and the number of shared therapeutic chemicals in diseases that do not share any genes. The biological process annotation of disease genes from the Gene Ontology can impart more comprehensive profiles of diseases than the disease genes alone.

Keywords: Disease similarity · Disease-gene functional annotation · Disease-related biological functions

1 Introduction

In the past two decades, methods based on the genes shared between diseases have been proposed to analyze disease relationships and disease similarity [1, 2]. Disease similarity measurement methods based on phenotypes, drugs, or disease-causing molecules are important for drawing connections between diseases, because these methods may lead to improved treatment and therapeutic plans. Moreover, drug repurposing benefits from disease similarity results, as many studies have shown that similar diseases can be treated by similar drugs [1]. Disease similarity measurements can be accomplished via networks and undirected graphs [2, 3], numeric value similarity scores [2, 3], or matrices [1]. The computation to estimate disease similarity can be based on genes, phenotypes, symptoms, etc., but comparing diseases strictly by their genes does not always reveal relationships between diseases. This is because only about 30% of genes with phenotype-causing mutations could be directly associated to more than one disease, while the remaining 70% of genes are responsible for at most one disease [4]. As a result, we propose a method that explores disease relationships by comparing diseases by their shared biological processes rather than their shared genes. The method relies on biological process (BP) weight to compare diseases that do not share any or some of their genes.

This raises questions about whether BP-weight scores could be used to measure the similarity between diseases that do not share any genes, reveal previously unknown therapeutic chemicals for diseases, or identify disease-causing biological processes.

2 Background and Related Work

Carson et al. proposed a matrix-based disease similarity technique that used the set of shared genes between every disease pair [1]. They calculated a gene uniqueness (GU) value for every gene based on how many diseases were associated to that gene. Then, the GU values were used to calculate disease similarity [1]. Jia et al. presented a standardized system called *eRAM* for rare diseases [2]. The *eRAM* system was developed as an encyclopedia involving around 10 million scientific publications explored with text-mining processes [2]. Further, the *eRAM* system utilized gene-based disease similarity, phenotype-disease, and symptom-disease relationship data [2]. Cheng et al. proposed and implemented a disease similarity approach that combined two techniques, one for functional similarity and the other for disease semantic similarity [3]. Another method called *PedAM* also used disease similarity for developing a database for pediatric disease annotations [5]. A disease similarity network was developed by Wei et al. based on an ontological method, machine learning, and concept embedding [6]. A comprehensive genome-scale study by Cornish et al. [7] was developed to analyze and reduce the bias of data availability of disease genes and highly studied genes. This is a common issue in most genetic-related research work [7]. An online disease similarity system called *DisSim* implemented various similarity measures for Disease Ontology (DO) terms [8]. Le et al. proposed a disease similarity method based on phenotype similarity networks [9].

The Online Mendelian Inheritance in Man (OMIM) database is a comprehensive source of human diseases and the genes associated to them [1, 10]. OMIM states that 3,036 out of the 4,364 genes with phenotype-causing mutations are associated to one phenotype [4]. This could mean there are 3,036 diseases that cannot be genetically compared to any other diseases because their genes are unique. This raises a need for a disease comparison method based on other disease attributes. In this work, we use the Gene Ontology (GO) to obtain the biological processes associated to disease genes [11]. We then use the Comparative Toxicogenomics Database (CTD) to find if diseases associated to the same biological processes could be treated using the same therapeutic chemicals. Currently, literature on disease-biological process relationships is relatively limited [6–9]. Therefore, our method shall extend our knowledge of relationships between diseases, biological process functions, and genes from GO.

3 Methods and Techniques

3.1 Gene Ontology and Biological Process Annotations

Computations that estimate similarity between diseases rely on attributes like disease genes, symptoms, phenotypes, etc. For example, let diseases d_1 and d_2 be represented as sets of genes so that $d_1 = \{g_1 g_2 g_3\}$ and $d_2 = \{g_1 g_4\}$. In this case, diseases d_1 and d_2 share gene g_1 . Gene-based methods can impart a similarity between diseases d_1 and d_2 by using a similarity function on the shared gene g_1 . Cheng et al. have a comprehensive study of various disease similarity measures and major databases of disease vocabulary and annotations [12].

In this work, we used BP functional annotation of disease genes. BP functional annotation is the process of associating BP functions to genes from the GO [11]. The gene-disease associations can be obtained from online resources such as OMIM, DO, or CTD [13–15]. We obtained the genes and annotated each gene with its BP functions. For example, let diseases x and y be represented as sets of genes so that $x = \{g_1 g_2\}$ and $y = \{g_3\}$. Then, each gene is represented as a set of BP functions. For example, let $g_1 = \{p_1 p_2\}$, $g_2 = \{p_3 p_4 p_5\}$ and $g_3 = \{p_1 p_2 p_6 p_7\}$. Finally, we substitute the genes with the corresponding BP functions to get $x = \{p_1 p_2 p_3 p_4 p_5\}$ and $y = \{p_1 p_2 p_6 p_7\}$. Now, diseases x and y are represented as sets of BP functions and share the BP terms $\{p_1 p_2\}$ despite not sharing any genes, so we introduce a weight function $W(p)$ that calculates the weight, or contribution, of every BP function p . The weight method is as follows:

$$W(p) = 1 - \sqrt{\frac{count_p}{n}} \quad (1)$$

where the weight W is a function of a biological process p , $count_p$ is the number of diseases annotated with the biological process p , and n is the total number of diseases in the data set [1].

When counting the number of diseases annotated with a biological process p , child and parent BP terms should be counted in the same way and without any distinction. This is because the count for parent terms is larger than the count for child terms, so the weight function W will yield smaller weight values (closer to 0) for broad parent terms and higher weight values (closer to 1) for specific child terms. This method is used in the similarity score method shown in Eq. (2), which finds the set of shared BPs between every disease and uses the hierarchical structure of the GO [11] to remove general parent terms and keep specialized child terms. For example, if diseases x and y were to share the terms “cellular process” and its child “cellular metabolic process”, we would only keep the child “cellular metabolic process” since the parent is redundant. BP-weight scores are calculated as follows:

$$S(x, y) = \sum_{p_s \in remove_ancestors(X \cap Y)} W(p_s) \quad (2)$$

where X and Y are the sets of biological processes associated with the corresponding diseases x and y , the function $remove_ancestors$ removes biological process parents, p_s is a biological process shared by x and y that is not the parent of any shared biological

processes, and $\sum W(p_s)$ is the summation of the weight of every shared biological process p_s . Note that the *remove_ancestors* function uses the GOATOOLS module [16] for Python and the Gene Ontology [11] to determine parent and child biological processes.

3.2 Biological Process Jaccard Score

In this research, we also used the biological process Jaccard score for every disease pair as a disease similarity metric. The BP Jaccard score for similarity between disease x and disease y is computed as follows:

$$J_{BP}(x, y) = \frac{|remove_ancestors(X \cap Y)|}{|remove_ancestors(X \cup Y) \cup remove_ancestors(X \cap Y)|} \quad (3)$$

where X and Y are the sets of biological processes associated to the corresponding diseases x and y , and the function *remove_ancestors* removes biological process parents. The equation removes redundant BP term ancestors from the nominator but does not remove shared BP terms from the denominator, even if they are redundant. This ensures that Eq. (3) can only return values between zero and one.

4 Evaluation and Results

The experiments focused on BP-weight scores from Eq. (2) because they were better predictors of shared therapeutic chemicals than the BP Jaccard scores from Eq. (3). The source code and data are available on <https://github.com/Luis-Licea/BP-Functional-Annotations>.

4.1 Experiment 1 - Phenotype Jaccard Score and BP-Weight Correlation

We created a similarity matrix of 5,415 diseases using the OMIM *MorbidMap* file [13], obtained the genes for each disease from the OMIM *GeneMap2* file [13], and annotated each gene with its biological processes using the *GOA_Human* file [11]. We compared each disease with the remaining 5,414 diseases based on the GU method proposed in [1]. We looked for the pairs with the highest GU scores and found 2,717 disease pairs with non-zero GU scores. The remaining 2,689 out of the 5,415 diseases had a GU similarity score of zero because they did not share genes with other diseases. For the 2,717 disease pairs with non-zero GU scores, we noticed that many pairs had the same similarity. For example, there were 1,316 pairs that had a GU similarity score of 0.98, so we examined some of these pairs with our BP-weight method. We calculated the weight of each biological process p using Eq. (1) and compared all the diseases based on their shared BPs using Eq. (2). We picked four pairs having a GU similarity score of 0.95, which can be seen in Table 1, and we found that BP-weights exhibited different similarity scores. We compared GU and BP-weight scores with phenotype Jaccard scores, and the results showed that BP-weight scores were weakly correlated with phenotype Jaccard scores, as shown in Table 1. However, since the correlation was nonlinear, we validated BP-weight using the number of shared chemicals in the following experiments.

Table 1. The table shows four disease pairs that have the same GU score, but different BP-weight scores. We used the number of associated and shared phenotypes to calculate the phenotype Jaccard (PJ) scores. We found a weak, positive correlation ($r = 0.37$) between PJ Scores and BP-weight. However, the fourth disease pair exhibited a higher BP-weight (93.05) than the first pair while having the same PJ score (1). This made the correlation nonlinear. Disease names and OMIM IDs are also given. The IDs can be used to find information about diseases in OMIM and CTD. We obtained the Phenotype Jaccard data from <http://ctdbase.org/tools/vennViewer.go>.

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	PJ	GU	BP-Weight
1	610140	176670	Heart-hand syndrome	Hutchinson-Gilford	1	0.95	13.98
2	615278	137215	Cardiofaciocutaneous	Gastric cancer	0.23	0.95	23.98
3	614592	101600	Bent bone dysplasia	Pfeiffer syndrome	0.75	0.95	74.57
4	151623	607107	Li-Fraumeni syndrome	Nasopharyngeal	1	0.95	93.05

4.2 Experiment 2 - Correlation Between Shared Chemicals and BP-Weight

In this experiment, we evaluated the similarity of four disease pairs (Table 2) having a GU score of zero. We obtained the BP-weight and number of shared chemicals for each disease pair, and noticed there was a very strong, positive correlation between BP-weight and the number of shared chemicals, as illustrated in Fig. 1.

Table 2. The table shows that as BP-weight increases, the number of shared chemicals (SC) increases.

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	SC	GU	BP-Weight
1	139210	611590	Myhre syndrome	Renal tubular acidosis	23	0	4.89
2	607822	613205	Alzheimer disease, type 3	Muscular dystrophy	34	0	14.37
3	608232	181405	Leukemia, Philadelphia	Scapuloperoneal spinal	63	0	17.62
4	104310	101600	Alzheimer disease 2	Pfeiffer syndrome	89	0	23.37

4.3 Experiment 3 - Comparing Diseases Without Genes in Common

In order to validate that BP-weight is a good predictor of shared chemicals, we chose two cancer-related diseases and two dementia-related diseases and created five disease pairs such that they shared no genes in common (Table 3). We then calculated the BP-weights

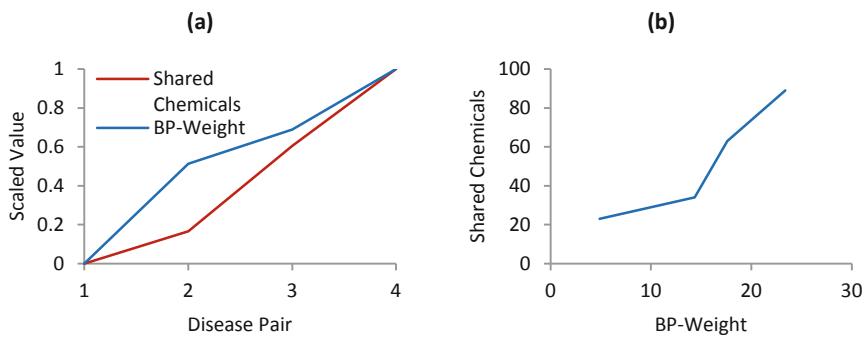


Fig. 1. Graphs (a) and (b) summarize the data from the four disease pairs in Table 2. Graph (a) shows that for each disease pair, The BP-weight and the number of shared chemicals increase while GU equals zero. Meanwhile, graph (b) shows a very strong, positive correlation ($r = 0.93$) between BP-weight and shared chemicals

for the five pairs. The data showed that *Alzheimer's* disease and *breast neoplasms* shared more therapeutic chemicals than *Alzheimer's* disease and *Lewy body* disease (compare the second and fifth disease pairs in Table 3). This was unexpected because *Alzheimer's* disease and *Lewy body* disease are types of dementia.

Table 3. Five disease pairs with a GU of zero and ordered by BP-weight. As BP-weight increases, the number of shared chemicals increases, as shown in the table and in Fig. 2.

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	SC	GU	BP-Weight
1	127750	176807	Lewy Body Disease	Prostatic Neoplasms	2,043	0	42.09
2	127750	104300	Lewy Body Disease	Alzheimer Disease	1,992	0	53.32
3	127750	114480	Lewy Body Disease	Breast Neoplasms	2,006	0	71.61
4	104300	176807	Alzheimer Disease	Prostatic Neoplasms	5,870	0	75.92
5	104300	114480	Alzheimer Disease	Breast Neoplasms	5,961	0	100.48

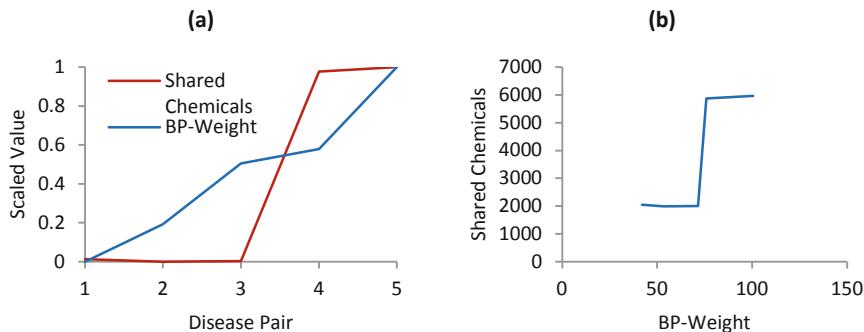


Fig. 2. Graph (a) shows the scaled BP-weight and scaled number of shared chemicals obtained by comparing five disease pairs having a GU of zero. Graph (b) shows a strong, positive correlation ($r = 0.80$) between BP-weight and shared chemicals.

4.4 Experiment 4 - Correlation Between Shared Chemicals and BP-Weight for a Given Disease

We demonstrated that BP-weight is positively correlated with shared therapeutic chemicals, and that diseases that do not share genes can share therapeutic chemicals (Fig. 2, Table 3). Therefore, we set up this evaluation to determine how well do higher BP-weights correlate to higher numbers of shared therapeutic chemicals for a given disease. We compared each disease in the data set with the remaining 5,414 diseases and sorted the pairs with the highest BP-weights in non-increasing order. We picked the first 10 disease pairs such that they had a GU value of zero, and were associated to the same disease, in this case *colorectal neoplasms* (Table 4). We then plotted the BP-weight and number of shared chemicals associated to each disease pair. After controlling for GU and one of the diseases in each disease pair (*colorectal neoplasms*), we found a very strong, positive correlation between BP-weight and shared chemicals (Fig. 3).

4.5 Experiment 5 - Comparing Diseases with Genes in Common

Using the same data set of 2,717 disease pairs used in Sect. 4.1, we sorted the pairs based on GU and picked the first highest 10 disease pairs such that they had the same GU. In this case, the 10 disease pairs had a GU of 0.98. We then calculated the BP-weight for each disease pair, as shown in Table 5. We found no correlation ($r = -0.02$) between BP-weight and shared chemicals (Fig. 4). This indicates that BP-weight is not a good predictor of shared chemicals whenever BP is low, or whenever one or more genes are being shared, since the GU score for every pair is 0.98.

4.6 Experiment 6 - Comparing Diseases with High BP-Weights

Using the similarity matrix from Sect. 4.1, we compared each disease with the remaining 5,414 diseases and stored the disease pairs with the highest BP-weight scores. This produced a file with 5,415 disease pairs with non-zero BP-weight scores. This means that every disease in the data set shared at least one BP term with another disease. Note

Table 4. Ten disease pairs with a GU of zero and ordered by BP-weight. As BP-weight increases, the number of shared chemicals increases, as shown in the table and in Fig. 3.

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	SC	GU	BP-Weight
1	608747	114500	Insulin-Like Growth	Colorectal Neoplasms	470	0	53.58
2	603933	114500	Microvascular	Colorectal Neoplasms	794	0	65.56
3	614816	114500	Loeys-Dietz Syndrome	Colorectal Neoplasms	548	0	66.37
4	613443	114500	Mental Retardation	Colorectal Neoplasms	119	0	67.68
5	114290	114500	Campomelic Dysplasia	Colorectal Neoplasms	185	0	69.59
6	146255	114500	Barakat syndrome	Colorectal Neoplasms	148	0	71.77
7	614962	114500	Leptin Deficiency	Colorectal Neoplasms	387	0	74
8	180700	114500	Robinow Syndrome	Colorectal Neoplasms	187	0	86.85
9	607785	114500	Leukemia	Colorectal Neoplasms	827	0	110.98
10	125853	114500	Diabetes Mellitus, Type 2	Colorectal Neoplasms	5,623	0	211.6

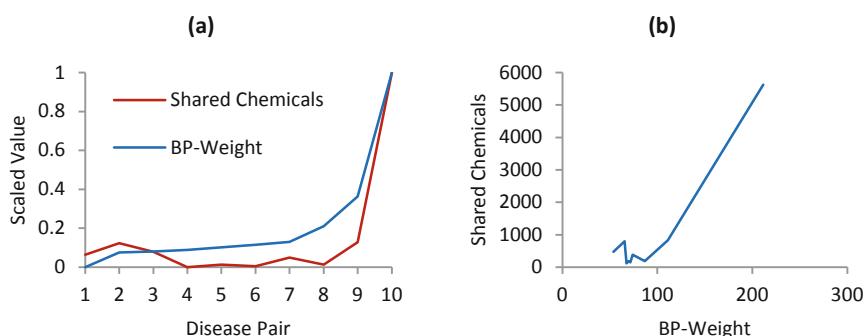


Fig. 3. Graph (a) shows the scaled BP-weight and scaled number of shared chemicals obtained by comparing 10 diseases to *colorectal neoplasms* while GU is zero. Graph (b) shows a very strong, positive correlation ($r = 0.95$) between BP-weight and shared chemicals. Note that five pairs form a cluster of BP-weights between 65 and 72.

Table 5. Ten disease pairs ordered by BP-weight and each with a GU of 0.98. The association is hard to interpret because the shared therapeutic chemicals could be associated exclusively to the shared genes or to the shared BP terms, and in this case, diseases share genes and BP terms.

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	SC	GU	BP-Weight
1	600204	614284	Epiphyseal 2	Stickler Syndrome, Type V	47	0.98	1.57
2	600969	603932	Epiphyseal 3	Intervertebral disc disease	43	0.98	2.57
3	134600	612718	Fanconi Syndrome 1	Arginine-Glycine	126	0.98	4.34
4	227810	125853	Fanconi Syndrome 4	Diabetes Mellitus, Type 2	485	0.98	5.46
5	228000	159950	Farber	Spinal Muscular Atrophy	115	0.98	8.1
6	208150	616325	Pena Shokeir	Myasthenic Syndrome	28	0.98	9.81
7	266200	102900	Pyruvate Kinase	Adenosine Triphosphate	175	0.98	9.95
8	601709	104300	Quebec platelet disorder	Alzheimer Disease	249	0.98	11.5
9	609054	114480	Fanconi Anemia	Breast Neoplasms	80	0.98	15.75
10	178600	265450	Hypertension	Pulmonary Veno	139	0.98	42.06

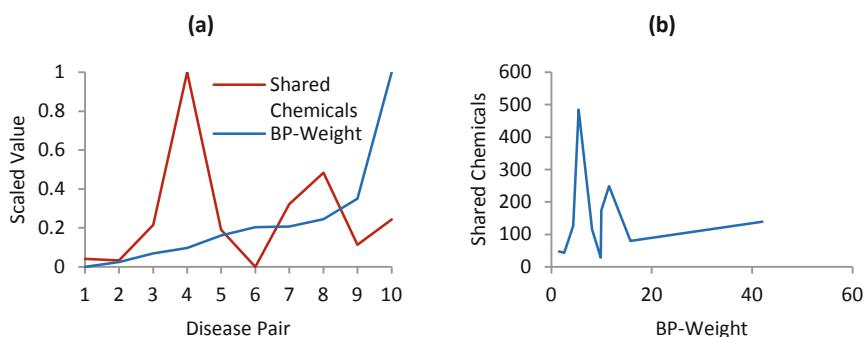


Fig. 4. Graph (a) shows the scaled BP-weight and scaled number of shared chemicals obtained by comparing 10 disease pairs while GU equals 0.98. Graph (b) shows no correlation ($r = -0.02$) between BP-weight and shared chemicals while controlling for GU. Note that the first nine pairs form a cluster of BP-weights between 1 and 16.

that disease pairs were allowed to share genes since they were simply chosen based on having the highest BP-weight scores. We sorted the 5,415 entries in non-increasing order based on BP-weight and randomly picked 30 disease pairs out of the 5,415. Out of these 30 disease pairs, 10 could not be used because CTD did not recognize the disease IDs. This left us with the 20 diseases shown in Table 6. We found a very strong, positive correlation between BP-weight and shared chemicals (Fig. 5).

Table 6. Twenty randomly selected disease pairs

Pair	OMIM ID 1	OMIM ID 2	Disease 1	Disease 2	SC	GU	BP-Weight
1	617115	607655	Peeling Skin Syndrome	Skin Fragility-Woolly	40	0	1.11
2	600204	607078	Epiphyseal dysplasia	Epiphyseal dysplasia	15	0	1.57
3	617072	128100	Myopathy	Dystonia musculorum	13	0	1.96
4	602772	268220	Retinitis Pigmentosa 25	Rhabdomyosarcoma	10	0	2.13
5	615837	616515	Deafness, Autosomal	Deafness, Autosomal	2	0	2.33
6	604286	177170	Limb-girdle muscular	Pseudoachondroplasia	12	0	2.4
7	609220	225400	Bruck syndrome 2	Ehlers-Danlos syndrome	25	0	2.47
8	615348	612229	Nemaline Myopathy 8	Colorectal Neoplasms	26	0	3.07
9	617280	608751	Atrial Fibrillation	Cardiomyopathy	22	0	3.59
10	115430	145680	Carpal Tunnel	Dystransthyretinemic	236	0.98	5.32
11	617388	615225	Autoinflammation	Palmoplantar	34	0.97	8.63
12	616313	616314	Myasthenic	Myasthenic Syndrome	120	0.98	8.89
13	616532	609423	Encephalitis, Herpes	HIV Infections	258	0	18.71
14	614810	142680	Multiple Sclerosis	Periodic fever, familial	290	0.98	21.37
15	614325	614332	Pitt-Hopkins-Like	Chromosome 2p16.3	93	0.98	23.26
16	122470	114500	De Lange Syndrome	Colorectal Neoplasms	237	0	30.41
17	186580	266600	Blau syndrome	Inflammatory Bowel	54	0.98	38.09
18	156250	607785	Metachondromatosis	Leukemia	114	0.97	50.91
19	616069	211980	Inflammatory Skin	Lung Neoplasms	577	0.98	74.41
20	151623	137800	Li-Fraumeni Syndrome	Glioma	1,228	0.95	93.05

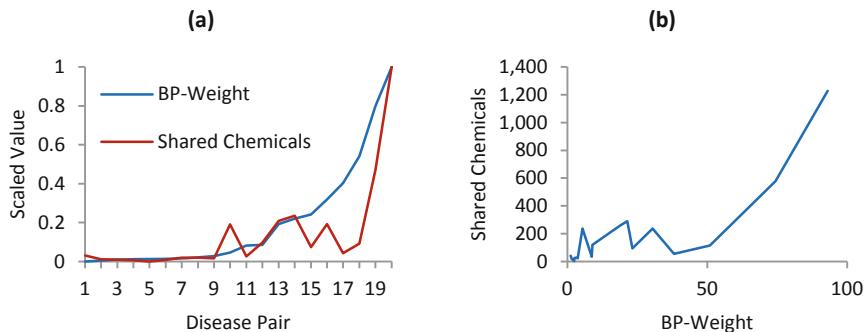


Fig. 5. Graph (a) shows the scaled BP-weight and scaled number of shared chemicals obtained from 20 randomly selected disease pairs. Graph (b) shows a very strong, positive correlation ($r = 0.85$) between BP-weight and shared chemicals. Note that the first 12 pairs form a cluster of BP-weights between 1 and 9.

5 Analysis of Results and Discussion

Disease similarity methods that only rely on disease genes can be good measures of similarity between diseases but can be improved in certain cases as shown and discussed in this paper. Our method relies on the functional profiles from the BP taxonomy of the GO. This methodology can measure the similarity between diseases with and without shared genes. The evaluation and experiment results of the presented method demonstrated a fairly impressive and encouraging performance compared with one of the recent methods that relies solely on disease genes. We also demonstrated the performance of the proposed BP-weight method. In most cases, the disease similarity results show good performance compared with shared chemicals as diseases that are more similar share more therapeutic chemicals. We showed that diseases that shared some or none of their genes still had some similarity. See Fig. 1, Fig. 2, Fig. 3, and Fig. 5. For example, in Table 3 and Fig. 2, we evaluated the similarity of diseases that did not share any genes with remarkably good and encouraging results. One of our evaluation experiments showed that there was no correlation ($r = -0.02$) between BP-weight and shared chemicals whenever we only compared diseases that shared genes. This means BP-weight is not a good predictor when comparing diseases that share genes. This raises the question, can diseases known to be similar have small BP-weights, or do the BP-weights seem low because the underlying BP terms have not been found? Furthermore, could large BP-weights and low numbers of shared chemicals be used to identify disease pairs that have not been researched thoroughly for shared chemicals? Hopefully future research will reveal how individual BP terms or groups of BP terms are associated to therapeutic chemicals, and how these associations can be used to improve the BP-weight method. Currently, the BP-weight method can be improved since the sample BP-weight and shared chemical distributions are positively skewed and do not follow normal distributions. Also, the variance in the number of shared chemicals increases as BP-weight increases because each BP term included in the BP-weight may be associated to a different number of therapeutic chemicals.

6 Conclusion

A new method for disease similarity is presented in this paper and is based on the functional profiles of diseases from the biological process taxonomy of the Gene Ontology. In this work, we found that the BP-weight measurement method can be a good predictor of shared chemicals for diseases that do not share any genes. In the evaluation of our proposed method, experiments in Sects. 4.2, 4.3 and 4.4 showed strong correlations between BP-weight and shared chemicals while comparing diseases with GU similarity values of zero, while the experiment in Sect. 4.5 showed no correlation when comparing diseases with a GU of 0.98. This suggests that BP-weight is a good predictor of shared chemicals, except for diseases that share genes.

References

1. Carson, M.B., Liu, C., Lu, Y., Jia, C., Lu, H.: A disease similarity matrix based on the uniqueness of shared genes. *BMC Med. Genomics* **10**, 27–32 (2017). <https://doi.org/10.1186/s12920-017-0265-2>
2. Jia, J., et al.: eRAM: encyclopedia of rare disease annotations for precision medicine. *Nucleic Acids Res.* **46**(D1), 937–943 (2018). <https://doi.org/10.1093/nar/gkx1062>
3. Cheng, L., Li, J., Peng, J., Peng, J., Wang, Y.: SemFunSim: a new method for measuring disease similarity by integrating semantic and gene functional association. *PLoS ONE* **9**(6), e99415 (2014). <https://doi.org/10.1371/journal.pone.0099415>
4. OMIM Gene Map Statistics. <https://www.omim.org/statistics/geneMap>. Accessed 19 Sept 2020
5. Jia, J., et al.: PedAM: a database for pediatric disease annotation and medicine. *Nucleic Acids Res.* **46**(D1), 977–983 (2018). <https://doi.org/10.1093/nar/gkx1049>
6. Wei, D., Kang, T., Pincus, H.A., Weng, C.: Construction of disease similarity networks using concept embedding and ontology. *Stud. Health Technol. Inform.* **264**, 442–446 (2019). <https://doi.org/10.3233/SHTI190260>
7. Cornish, A.J., David, A., Sternberg, M.J.E.: PhenoRank: reducing study bias in gene prioritization through simulation. *Bioinformatics* **34**(12), 2087–2095 (2018). <https://doi.org/10.1093/bioinformatics/bty028>
8. Cheng, L., et al.: DisSim: an online system for exploring significant similar diseases and exhibiting potential therapeutic drugs. *Sci. Rep.* **6**, 1–6 (2016). <https://doi.org/10.1038/sre p30024>
9. Le, D.-H., Dang, V.-T.: Ontology-based disease similarity network for disease gene prediction. *Vietnam J. Comput. Sci.* **3**(3), 197–205 (2016). <https://doi.org/10.1007/s40595-016-0063-3>
10. About Online Mendelian Inheritance in Man. <https://www.omim.org/about>. Accessed 19 Sept 2020
11. Gene Ontology Annotation Downloads. <https://www.ebi.ac.uk/GOA/downloads>. Accessed 04 Apr 2020
12. Cheng, L., et al.: Computational methods for identifying similar diseases. *Mol. Therapy - Nucleic Acids* **18**, 590–604 (2019). <https://doi.org/10.1016/j.omtn.2019.09.019>
13. OMIM Data Downloads. <https://www.omim.org/downloads>. Accessed 19 Sept 2020
14. Disease Ontology Downloads. <https://disease-ontology.org/downloads>. Accessed 16 Sept 2020
15. Davis, A.P., et al.: The comparative toxicogenomics database: update 2017. *Nucleic Acids Res.* **45**(D1), 972–978 (2017). <https://doi.org/10.1093/nar/gkw838>
16. Klopfenstein, D., et al.: GOATOOLS: a python library for gene ontology analyses. *Sci. Rep.* **8**(1), 1–17 (2018). <https://doi.org/10.1038/s41598-018-28948-z>



Impact of Types of Change on Software Defect Prediction

Atakan Erdem^(✉)

University of Calgary, Calgary, AB, Canada
atakan.erdem1@ucalgary.ca

Abstract. Churn metrics are commonly used in software defect prediction due to high performance, language independence and ease of extraction. The main data sources for churn metrics-based prediction models in the literature are the change logs. But changes may also be due to several reasons other than fixing defects, such as enhancement and applying new requirements. Therefore, without awareness of type of the change and effects on defect prediction, fitting the best model is too difficult. In this paper, we propose a churn type-aware defect prediction model. We observe the impacts of churn-type awareness on prediction accuracy level. In our experiments, we used a real work item and change log dataset which is produced by a software development team. In our study, we also reveal the correlations among the change types.

Keywords: Churn metric · Change type · Defect prediction

1 Introduction

The most common type of changes in a software system is defect type changes. This means that most of the effort spent is due to defect fixing and also points to the biggest cost item of a software. With this respect defect prediction is one of the most focused research topics in software engineering and subject of many previous studies. These studies typically produce defect prediction models which allow software engineers to focus development activities on defect-prone code, thereby improving software quality and making better use of resources.

Software defect prediction can be formulated as a learning problem in software engineering, which has drawn growing interest from both academia and industry. Static code attributes are extracted from previous releases of software with the log files of defects and used to build models to predict defective modules for the next release. It helps to locate parts of the software that are more likely to contain defects.

The main three research questions addressed by most of the previous studies are [1]: How does context affect defect prediction? Which independent variables should be included in defect prediction models? Which modelling techniques perform best when used in defect prediction?

In this paper, we study on the effects of change types on defect prediction. Therefore, we would like to answer the question of “What is the impact of types of changes (churn) on defect proneness of the software?”.

Code churn is a measure of the amount of code change taking place within a software unit over time. It is easily extracted from a system's change history, as recorded automatically by a version control system. Most version control systems use a file comparison utility (such as diff) to automatically estimate how many lines were added, deleted and changed by a programmer to create a new version of a file from an old version. These differences are the basis of churn measures. In many previous researches, churn metrics are used for modelling defect prediction. For example in [2], churn metrics are used to predict system defect density. But changes on software are not only done due to defects but also due to some other reasons such as routine tasks during development or enhancement.

In this paper, correlations among change types and their effects on defect prediction are observed by conducting some experiments. As experiment dataset, we use a dataset consists of work items which are defined and assigned during development life cycle of a commercial issue tracking system. The work items in this dataset are recorded according to types of change on source code. We observe during the experiments that building a defect prediction model with the awareness of change type brings a distinguished improvement on prediction performance.

The organization of this paper is as follows. Section 2 describes the related work. Section 3 explains data collection. Section 4 presents the experiments and the observed results. Section 5 discusses our conclusions and future work.

2 Related Work

Researchers have developed methods and tools to better cope with software maintenance and evolution. Some approaches, use source code metrics to train prediction models, which can guide developers towards the change-prone parts of a software system. The main motivation for these approaches is that developers can better focus on these change-prone parts to take appropriate counter measures to minimize the number of future changes [3–6].

Other approaches, support developers in modification tasks that affect different source code locations by automatically eliciting past changes and change couplings between these source code entities. Moreover, the sensitivity to which the design of a system reacts to changes can be an indicator for its quality [7–9].

But none of these proposed solutions are designed being aware of type of changes. Without considering for which purpose the source code is changed, it's impossible to build an optimum model. For example in [10], information provided by change history of source codes is used in predicting defect-prone files. But since all changes in history are assumed to be done for bug fixing by excluding all other type of changes, the proposed accuracy level is not optimal.

With respect to the importance of knowing change types, in [11], a model is proposed to predict categories of changes.

In another interesting study [12], as an observation result of research question, "What factors contribute the bug fixing time delays most", it is claimed that one of the most influential factors is the sum of code churns. This points the importance of churn metrics

in the context of bug fixing time estimation. But if parameters of bug fixing time prediction models are fine-tuned by considering change types, performance of a bug fixing time prediction model is improved as well.

To the best of our knowledge, there's no proposed defect prediction model which is built using change type information. With this respect, our study is the only research that explores the effects of change types on defect prediction which is crucial to build an optimum prediction model.

3 Data Collection

Our experiments are based on work item data of a software development team which is responsible to develop a commercial issue tracking system. The interesting thing is that, they use their own software for issue tracking purposes. The database consists of data of *changes* and *work items* assigned to developer team members. We first extract all data in *json* format and convert it to relations of PostgreSQL database. The extracted data covers all the work item data from 2005 to 2015. The *work item* records include type of changes. There are mainly three change types defined: *Task*, *Defect* and *Enhancement*. Change types also can be defined by end users dynamically which may be different from the three types. In the *change* relevant relations, the details of change operations such as state of changes and related work items are stored.

After construction the relational model, in the context of data collection, we first extract the churn data of source files and then construct the churn metrics based on the extracted churn data.

3.1 Churn Data Extraction

The churn data extraction program is developed in python. The program simply gets the source code files in Java as input and returns the churn values. Getting these churn values, they are associated with change and the work item data in the database. In this way, the types of change of whole code chucks are explored.

3.2 Churn Metrics Construction

After extracting churn data with change type information, we construct the metrics which are used as inputs in our experiments. In Table 1, the churn metrics are given.

Metric values are generated by utilizing the extracted churn data and the change and work item database.

Table 1. Churn metrics.

Cyclometric complexity	Halstead programming effort
Decision density	Halstead Programming Time
Essential density	Halstead Volume
Branch count	Maintenance Severity
Condition count	lines_added_task
Cyclometric density	lines_removed_task
Decision count	edit_frequency_task
LOC	people_task
Total operands	lines_added_defect
Total operators	lines_removed_defect
Unique operands count	edit_frequency_defect
Unique operators count	people_defect
Halstead difficulty	pre
Halstead length	Essential Complexity
Halstead level	Path

4 Experiments and Results

Though there are three major types of change (task, enhancement and defect), the churn data we extract only associated with two types of change (task and defect) in the change and work item database. Thus, we use two types of change in the experiments. The aim of the experiments is to observe the effects of change type information on prediction performance. For this purpose, we conduct two sets of experiments. In the first set, we observe the effects of various combinations of change types on prediction performance. In the second set, first we select the most informative churn metrics and then apply prediction algorithms to observe the prediction accuracy.

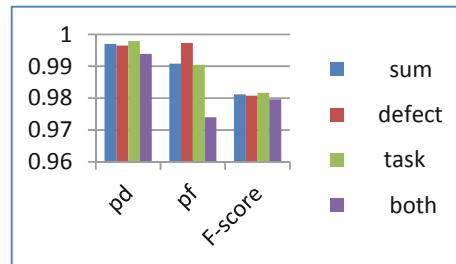
4.1 Experiment I

In Experiment I, we define four different scenarios. These are: i) combined churn metrics (sum of tasks and defects), ii) only defect based churn metrics, iii) only task based churn metrics and iv) both categories of churn metrics (both tasks and defects). For each scenario, we apply widely used and well-known prediction algorithms. These are: Logistic Regression (LR), Naïve Bayes (NB), LR/SMOTE, NB/SMOTE, LR/cost sensitive classifier (CSC), NB/CSC, J48/SMOTE, Random Forest (RF)/SMOTE. Each algorithm is repeated 100 times and 10-fold cross validation is applied. In Tables 2, 3, 4, 5, 6, 7, 8 and 9 and in Figs. 1, 2, 3, 4, 5, 6, 7 and 8 the results of the experiments according to various combinations of change types are given where pd: True/Positive rate, pf: False/Positive rate.

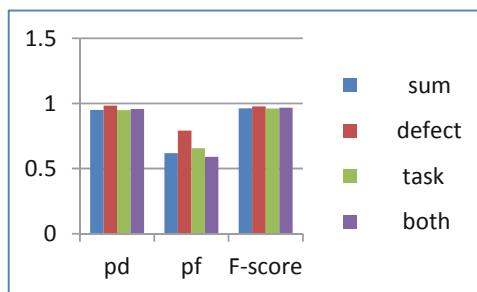
As it is seen in the tables and the figures, for 3 out of 4 metrics (sum, task and both) decision tree and random forest helps improve the accuracy of prediction model such

Table 2. Logistic regression experiments.

LR	Sum	Defect	Task	Both
pd	0,997	0,9965	0,9979	0,9939
pf	0,9908	0,9973	0,9905	0,974
F-score	0,9812	0,9808	0,9817	0,9796

**Fig. 1.** Graph of Logistic Regression experiments**Table 3.** Naïve Bayes experiments

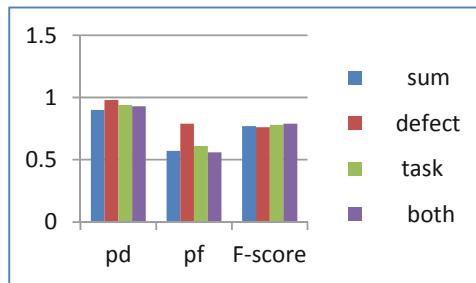
NB	Sum	Defect	Task	Both
pd	0,9497	0,9837	0,9489	0,957
pf	0,6183	0,7918	0,656	0,5906
F-score	0,9631	0,9779	0,9621	0,9675

**Fig. 2.** Graph of Naïve Bayes experiments

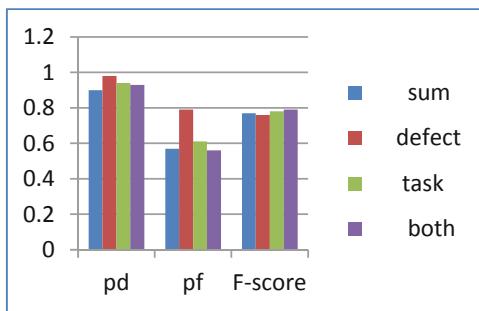
that pf is much lower than NB and LR, and pd is still high. According to this experiment, using J48 and RF return better prediction results than NB and LR. And using defect change type instead of whole types of changes (or task), prediction accuracy is improved significantly.

Table 4. LR/SMOTE experiments

LR/SMOTE	Sum	Defect	Task	Both
pd	0,9	0,98	0,94	0,93
pf	0,57	0,79	0,61	0,56
F-score	0,77	0,76	0,78	0,79

**Fig. 3.** Graph of LR/SMOTE experiments**Table 5.** NB/SMOTE experiments

NB/SMOTE	Sum	Defect	Task	Both
pd	0,94	0,98	0,93	0,93
pf	0,59	0,79	0,6	0,55
F-score	0,78	0,76	0,78	0,8

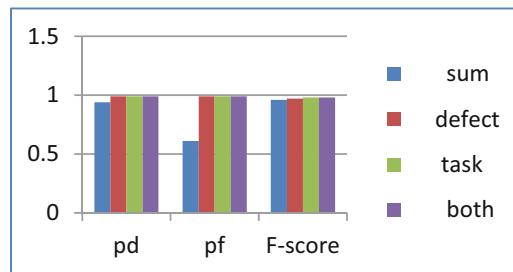
**Fig. 4.** Graph of NB/SMOTE experiments

4.2 Experiment II

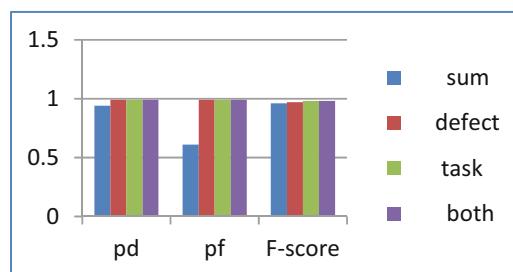
In the second experiment we make correlation analysis among churn metrics on defects. In the first attempt, we get the results in Table 10.

Table 6. LR/cost sensitive classifier experiments

LR/cost sensitive	Sum	Defect	Task	Both
pd	0,94	0,99	0,99	0,99
pf	0,61	0,99	0,99	0,99
F-score	0,96	0,97	0,98	0,98

**Fig. 5.** Graph of LR/cost sensitive classifier experiments**Table 7.** NB/cost sensitive classifier experiments

NB/cost sensitive	Sum	Defect	Task	Both
pd	0,94	0,98	0,94	0,94
pf	0,59	0,79	0,65	0,61
F-score	0,78	0,97	0,96	0,96

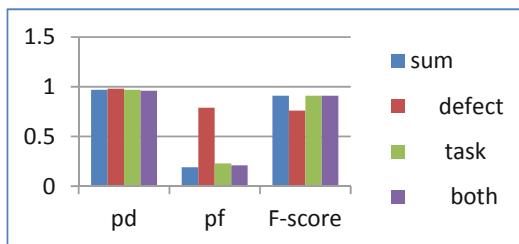
**Fig. 6.** Graph of NB/cost sensitive classifier experiments

As it is seen in Table 10, the highest correlation is between *lines_added_defect* and response variable and also *people_defect* and response variable.

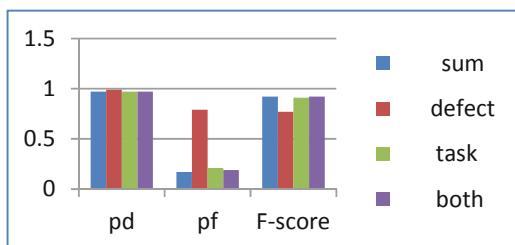
After the first attempt, we make further analysis of feature selectin. For this purpose, we use three techniques: i) Information Gain, ii) Gain Ratio and iii) Wilcoxon rank test.

Table 8. J48/SMOTE experiments

J48/SMOTE	Sum	Defect	Task	Both
pd	0,97	0,98	0,97	0,96
pf	0,19	0,79	0,23	0,21
F-score	0,91	0,76	0,91	0,91

**Fig. 7.** Graph of J48/SMOTE experiments**Table 9.** RF/SMOTE experiments

RF/SMOTE	Sum	Defect	Task	Both
pd	0,97	0,99	0,97	0,97
pf	0,17	0,79	0,21	0,19
F-score	0,92	0,77	0,91	0,92

**Fig. 8.** Graph of RF/SMOTE experiments

Information Gain

After applying feature selection using Information Gain, we get the ranking in Table 11.

Selected attributes: 3, 4, 1, 2, 8, 7, 5, 6: 8

We choose the first four attribute (3, 4, 1, 2) and then use Naive Bayes to predict post defects. The filtering causes both pf and pd to be reduced (compare to origin NB).

pd : 0.70, pd : 0.37

Table 10. Correlation analysis among churn metrics

lines_added_task	0.12777599
lines_removed_task	0.11111003
edit_frequency_task	0.20081626
people_task	0.16895651
lines_added_defect	0.21568087
lines_removed_defect	0.08929775
edit_frequency_defect	0.19620176
people_defect	0.21284774

Table 11. Feature selection using information gain

0.494	3 edit_frequency_task
0.337	4 people_task
0.273	1 lines_added_task
0.25	2 lines_removed_task
0.168	8 people_defect
0.168	7 edit_frequency_defect
0.148	5 lines_added_defect
0.105	6 lines_removed_defect

Gain Ratio

After applying feature selection using Gain Ratio, we get the ranking in Table 12.

Table 12. Feature selection using gain ratio

0.252	8 people_defect
0.252	7 edit_frequency_defect
0.244	5 lines_added_defect
0.176	3 edit_frequency_task
0.175	6 lines_removed_defect
0.17	4 people_task
0.163	2 lines_removed_task
0.154	1 lines_added_task

Selected attributes: 8, 7, 5, 3, 6, 4, 2, 1: 8

We choose the first four (8, 7, 5, 3) and again applied Naive Bayes to predict post defects. The filtering causes both pf and pd to be reduced (compared to originNB).

pd: 0.72, pf = 0.35.

According to above result, decision tree and random forest still outperform NB + feature selection technique.

Wilcoxon Ranking Test

For this purpose, we write a script in python to do a pairwise comparison of the performance of J48 and random forest for each metrics. Here is the metrics which significantly outperform others:

pf: defect based churn metrics

pd: combined churn metrics

F-measure: combined churn metrics

As defect based churn metric performed well for pf, we also check the balance as well and in the case of balance again combined churn metrics have the highest accuracy. The reason is that pf for defect based is slightly higher than others and pf for defect based is poor. Balance is a weighted average of pd and pf, therefore, for combined metrics it has the highest value.

In conclusion of the experiments, we can advise that the combination of both task and defect-based metrics for defect prediction modelling should be used.

5 Conclusion

In this paper, we propose an alternative defect prediction approach which is based on change types. We believe that being aware of change type, accuracy level of defect prediction models can be improved. For this purpose, we define churn metrics based on change types and apply well-known prediction algorithms. The experiments show that only J48 and Random Forest algorithms are sensitive to change types and improve prediction performance.

On the other hand, we also do the same experiments after selecting the most informative metrics using three feature selection methods. At the end of these experiments, we see that using both task and defect-based metrics for defect prediction, improve the prediction performance.

As a future work, we aim to do the same experiments on various and big databases.

References

1. Hall, T., Beechman, S., Bowes, D., Gray, D., Counsell, S.: A systematic literature review on fault prediction performance in software engineering. In: IEEE Transactions on Software Engineering, vol. 38, no. 6, November/December 2012
2. Nagappan, N., Ball, T.: Use of relative code churn measures to predict system defect density. In: ICSE 2005, St. Louis, MO, USA 15–21, May 2005

3. Li, W., Henry, S.: Object-oriented metrics that predict maintainability. *J. Syst. Softw.* **23**(2), 111–122 (1993)
4. Dagpinar, M., Jahnke, J.: Predicting maintainability with object-oriented metrics - an empirical comparison. In: Proceedings of Working Conference on Reverse Engineering, pp. 155–164 (2003)
5. Arisholm, E., Briand, L., Foyen, A.: Dynamic coupling measurement for object-oriented software. *IEEE Trans. Softw. Eng.* **30**(8), 491–506 (2004)
6. Girba, T., Ducasse, S., Lanza, M.: Yesterday's weather: guiding early reverse engineering efforts by summarizing the evolution of changes. In: Proceeding of International Conference on Software Maintenance, pp. 40–49 (2004)
7. Zimmermann, T., Weisgerber, P., Diehl, S., Zeller, A.: Mining version histories to guide software changes. In: Proceedings of International Conference on Software Engineering, pp. 563–572 (2004)
8. Ying, A., Murphy, G., Ng, R., Chu-Carroll, M.: Predicting source code changes by mining change history. *IEEE Trans. Softw. Eng.* **30**(9), 574–586 (2004)
9. Robbes, R., Pollet, D., Lanza, M.: Logical coupling based on fine-grained change information. In: Proceeding on Working Conference on Reverse Engineering, pp. 42–46 (2008)
10. Tsantalis, N., Chatzigeorgiou, A., Stephanides, G.: Predicting the probability of change in object-oriented systems. *IEEE Trans. Softw. Eng.* **31**(7), 601–614 (2005)
11. Giger, E., Pinzger, M., Gall, H.C.: Can We Predict Types of Code Changes? An Empirical Analysis, MSR 2012, Zurich, Switzerland (2012)
12. Ye, X., Bunescu, R., Liu, C.: Learning to rank relevant files for bug reports using domain knowledge. In: SIGSOFT/FSE 2014, Hong Kong, China, 16–22 November 2014



Analyzing Music Genre Popularity

Jose Fossi, Adam Dzwonkowski, and Salem Othman^(✉)

Wentworth Institute of Technology, Boston, MA 02115, USA

othmans1@wit.edu

Abstract. Music and the culture surrounding it are in perpetual states of change. With such a wide variety of genres continuously overtaking each other in popularity, there is a constant and inherent questioning of which genre is the most popular at any given time. Although billboards help give an overall understanding of what songs are presently dominating, they may not provide a full picture of what all people listen to daily, only scratching the surface. Social media in particular has dramatically increased the debates over music genre popularity and is a tool that can be used to gain insight on what forms of music people listen to in their daily lives that may not appear on the charts. The research in this paper focuses on utilizing social media as such a tool to perform analysis on what genres dominate the music industry today, both in an attempt to validate or find flaws in the billboards as well as understand what affects the popularity of genres as it stands. The analysis performed found both expected as well as contradicting results that answer several questions regarding the popularity of music genres today.

Keywords: Social media · PySpark · Music genres · Popularity

1 Introduction

The popularity of music genres has changed constantly throughout the years. With genres like country being at its high in the 40s, and blues shooting up in popularity in the 60s, it is evident that the majority of society's taste will continue to change. In this research, the data collected from various social media platforms is used to find which genres have been gaining or losing popularity based on how often they are mentioned on social media. How do the results obtained from social media compare to billboard charts? What genres are the most listened to in peoples' daily lives?

The motivation behind choosing music genre popularity as a topic stems from the fact that rap and hip-hop appear to be heavily dominating the music industry, and yet so many people listen to and prefer different genres. Although billboards (e.g. billboard.com, Rolling Stone Charts, Record World, etc.) generate lists of the “best” songs at the time, the data can be very subjective since it does not gather data from every music streaming platform. Conversely, the data collected through social media platforms is the culmination of users from multiple music streaming platforms, including some obscure platforms (e.g. Bot Libre!, Deezer, 8tracks, etc.) top billboards might not gather data from. Through graphing the data collected from the various platforms, it is possible to find the “taste” of music shifting over time, as well as comparing how different cultures

may have a different general taste when compared to another culture/group. With this initial motivation in mind, the research performed in this paper hopes to answer questions regarding the popularity of music genres today and how they compare amongst each other, as well as investigate the true validity of music popularity defined by the billboards.

In the remainder of this paper, related work is first discussed in Sect. 2 and an overview of the data is then given in Sect. 3. Section 4 explains the methodologies used to perform the research using the finalized dataset, and Sect. 5 then discusses the results of that research using graphs. Finally, the conclusion is presented in Sect. 6, along with both reflections and plans for future work.

2 Related Work

Music is one of the principal sources of entertainment and expression in human culture that has been around for thousands of years, and the variety within music makes the comparison of popularity of genres quite popular itself. Not only do readily available billboards and top-charts track what songs are most popular daily, but many studies have also been performed that analyze the changes in what genres are in highest demand over time. One such study, for example, analyzed the popularity of genres in the United States in 2018, finding that “the most popular genre among Americans was rock music” [5], with pop music a close second. The study was performed primarily using billboard charts and surveys. A different sample study looked more into the evolution of music over time in the U.S. rather than for a single year and visualized the shift in popularity amongst genres, similar in concept to the initial motivation behind the research in this paper. Also using billboard and top-charts as its source, the study found interesting data that showed how music genres move back and forth in popularity over time. One of the most interesting pieces of the study was the incredible observation it made regarding David Bowie’s popularity in particular, stating that “Back in the summer of 1983, David Bowie was enjoying a spell of incredible commercial success. ... Fast forward to 2016 and David Bowie is once again on the charts” [2]. The popularity of this individual artist in very different time periods is clear proof that tastes in music change dramatically over time, and is extremely relevant to the work being performed in this paper.

However, the execution of these relevant studies is quite different to the research at hand, making the research performed in this paper unique. While other studies have gathered data using billboards and top charts from various sources, this study is utilizing social media mentions as the source of data and avoids the billboards. This source of uniqueness is important and will hopefully provide a different angle through which results can be obtained.

What makes the problem at hand even more complex and captivating is that music is now deep in the digital world and has been taken over by streaming, a completely different method of consumption from the direct record and album purchases more popular in the past. Billboards and other top-charts would use album shipments and purchases to quantify the popularity of certain tracks, genres, and artists. Today, services such as Spotify, Pandora, Apple Music, and even YouTube are being used daily by millions of consumers playing songs on loop, and the charts need to adapt to this change which could

impact the accuracy of trends. In an article written by the New York Times referencing YouTube Music in particular, “the addition of billions of streams each week could have a substantial impact [on the Billboard chart]” [4]. Billboard’s decision when it comes to streams is that “1,250 clicks from a paying subscriber—or 3,750 clicks from a nonpaying user—are counted as the equivalent of one album sale” [4]. Although it makes sense to attempt to offset the ability to continuously stream tracks on repeat, the potential for inaccuracy, as well as the fact that paying subscribers have different impacts on the chart counts than non-paying users, appear to be quite volatile and could lead to inconsistency.

What’s more is that artists and record labels release songs and albums in such a way that they top the charts more easily and get higher numbers that they might merit. Artists are continuously breaking each other’s records in small amounts of time, raising further doubts about the accuracy of billboard charts. According to an article written by the Washington Post, Billboard’s senior vice president of charts and data development Silvio Pietroluongo said, “when streaming started, the idea was people would pick the tracks they wanted to hear, but now they’re being fed songs like a jukebox”, and the article continued to declare that “some artists appear to be gaming the system” [1].

With prior studies having not used social media but rather billboard charts as a data source for such research in the past, and with the dubious accuracy of those billboard top-charts, this paper hopes to find flaws in the billboard’s representation of music genre popularity using social media as a more accurate source. Although social media mentions can be quite volatile at times, it is believed that they will provide a better, more representative source of data for the problem at hand in this extensively digitalized and connected world.

3 Data Description and Collection

3.1 Gathering Data

The initial plan to gather data was to do so from both Twitter and Reddit, since these are two of the most prominent social media platforms that could potentially offer simple and useful gathering of such data. Other platforms, such as Facebook, were not seen as potential candidates since they did not appear to provide simple means of acquiring the data necessary for research, particularly since there are no hashtags that can be used as a point of reference. However, it was quickly found that Reddit would not be a viable option for this task either, since only moderators of the subreddits on the platform are allowed to pull its data. As a result, Twitter was the main social media platform through which data was gathered, using hashtags in tweets as the key parameters for finding data. Whenever a specified hashtag is encountered, that tweet is registered and stored as data.

After experiencing issues in acquiring a data gathering tool that provided the data necessary for the datasets (which will be discussed further in Sect. 3.3 of the paper), a free app was finally settled upon as the tool of choice for acquiring data from social media. The app, created by an unknown developer named Martin Hawksey, is called TAGS (Twitter Archiving Google Sheet) [3] and runs on the Google platform, allowing users to collect data from Twitter and store it in a Google sheet.

The app is very simple to use for the most part, only requiring quick setup before allowing it to act entirely independently. Upon “downloading” the app, a specially formatted Google sheet is opened that contains two tabs. The first is the information tab, which allows the user to input the fields required to run the script that collects the data, including the desired hashtag(s) to track, the time span for the search, and the maximum number of tweets to acquire. The second is the archive tab, which stores the gathered data in XLS-style format. Once all data has been entered in the information tab, the script requires both Twitter and Google account access to run. When access has been given, the script will run in the backend and gather the data for the specified hashtag up to the given time span, storing the results in the archive tab. After running the script for the first time, the sheet can be set to update automatically every hour so it does not need to be run manually for new data, allowing it to run independently for an extended period of time and accumulate large amounts of data.

Since a single Google sheet using TAGS stores all data in a single archive tab, each genre being tracked required its own separate Google sheet. In total, seventeen genres were tracked for research including: pop, metal, rock, blues, classical, country, dubstep, EDM, folk, gospel, hiphop, indie, instrumental, jazz, opera, rap, and reggae. An eighteenth mention was tracked as well- #music- as a sort of benchmark to compare against the genres. The archives for each of these sheets were set to update automatically and allowed to run for over a week. Once all archives were filled with sufficient data, each one was downloaded individually as a CSV file and stored locally to be cleaned, formatted, and used for research.

3.2 Cleaning and Formatting Data

After downloading each genre’s archive from the corresponding Google sheets onto a local machine, it was clear to see that some of the data included in the archive was improperly formatted for the research that had to be done on it. As a result, a Python script to clean and format the data was created. The script takes in several arguments, including the input file to clean and format, the filename to output the new data to, and the genre of the archive at hand. The cleaning that takes place involves removing several unnecessary columns in the CSV and ensuring proper UTF-8 encoding, while the formatting that occurs includes adding a new column in the CSV for the specified genre as well as ensuring proper values for the data by stripping out particular portions of the values. For example, the source platform for a posting is stored within an HTML tag, but the tag itself was not desired for research. The formatting and cleaning script was therefore in charge of ensuring the source was retrieved and stripped from the HTML tag. It is also important to note that it was necessary to clean out all commas from the data, since without doing so, splitting the data by comma in PySpark [6] would have been impossible and thrown off all list indices within the RDDs (Resilient Distributed Datasets). Although the size of the data decreased slightly as a result of cleaning the data, it was necessary to ensure the dataset being used contained information that could be translated into graphs without including any noise like random characters or null entries in the data.

Once all archives were cleaned and stored in a new file, it was necessary to combine them into one large, centralized file that could easily be read in PySpark rather than use

multiple separate files, which would have been much more difficult to work with since each cleaned file per genre would have had to be uploaded individually. This was done using a second Python script that easily combined given CSV files. The script takes in only two arguments to run, including a list of all the CSV files to combine and the filename to output the resulting combined data to. Due to the volume of data taken from TAGS, this script took about a minute to combine the data. After terminating, a fully cleaned, formatted, and combined CSV data file was ready to be used for analysis using PySpark code. The dataset included eleven columns: the ID, the text contained in the posting, the user/author, the exact time of publishing, the date, the platform on which the posting was published, the total user follower count, the total user friend count, the user location, the map of entities, and the genre.

4 Data Analysis

Analysis on the acquired data was performed using PySpark running in Google Colaboratory. Although Amazon S3 and WSL (Windows Subsystem for Linux) were viable options through which to run PySpark code, S3 proved to be too burdensome and problematic and WSL did not work well with the results and graphs produced by the code. The size of the data being used was not large enough to cause an issue when run through Colaboratory, and the graphs produced looked much better than other alternatives.

PySpark was downloaded into the Colaboratory environment by pulling and extracting the downloadable.tgz file from the Apache website, setting the proper environment variables, and creating both the SparkContext and SparkSession variables. With PySpark installed in the environment, the data file was uploaded. This process took a while due to the sheer size of the file, but after it was uploaded it could be read into an RDD as needed without issue. Upon being read, the data was immediately split by comma and the header was filtered out. The default number of partitions was set to two. It was decided that RDDs would be used over DataFrames for simpler and more effective analysis, since it was more familiar than DataFrames.

With the data loaded into the distributed file system and stored across the executors, the code was ready to be written for evaluation. Having had direct access to both the archives and the formatting/cleaning process for the data, there was no need to be familiarized with the data or run initial tests. Although doing so is typically a good idea, it was unnecessary for the situation at hand. No issues were encountered with the execution of the code and PySpark ran perfectly, mainly due to proper cleaning of the CSV file before it was uploaded. The only minor obstacle was to upload the data file every time the Colaboratory was opened for a fresh environment, since doing so took a while. Nevertheless, Colaboratory was only used for the visualization of the results and graphs. The downloadable Python file was the most important part, since it could be run wherever PySpark was installed including WSL through a spark-submit job.

5 Results and Evaluation

The results obtained matched expectations for the most part. The figure below (Fig. 1) shows the sorted total popularity by genre as found using the obtained data, with hip hop

coming out as a comfortable number one followed by rock, rap, and EDM. Interestingly, jazz music appeared to be more popular than country music, a finding that was not entirely expected.

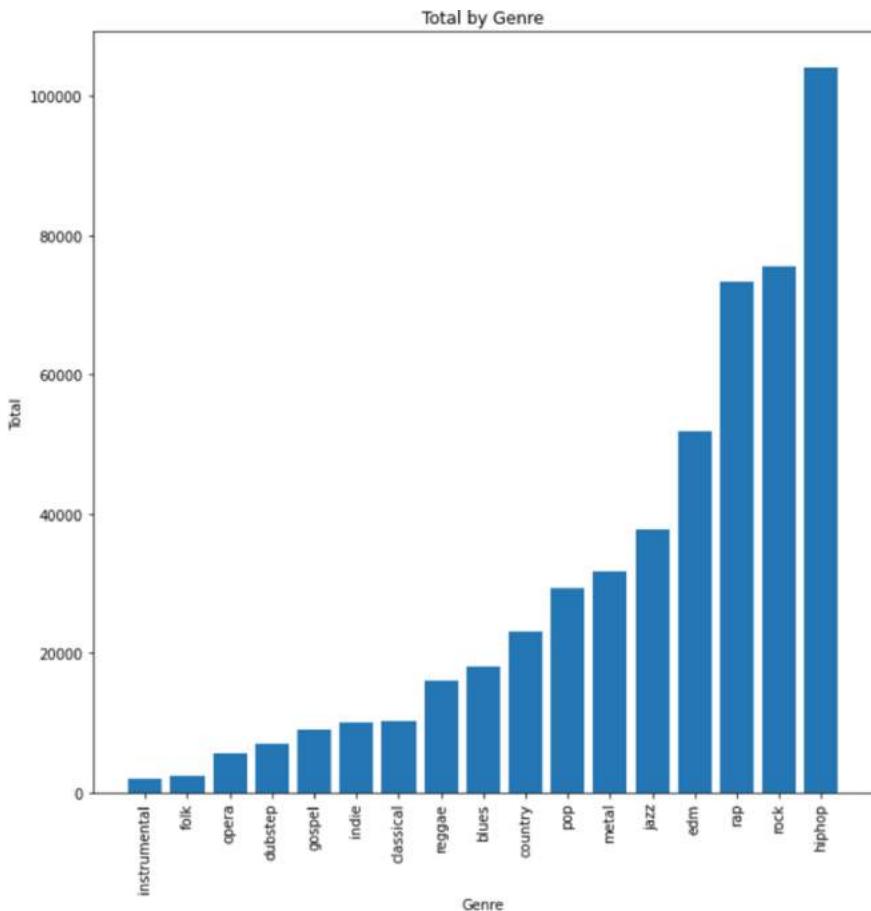


Fig. 1. Total popularity per genre, sorted in ascending order.

The distinct users (Fig. 2.) and locations (Fig. 3.) for the individual postings in the data were also analyzed in an attempt to understand the spread of unique users per genre. It was interesting to see that the less popular genres had a higher percentage of distinct users and locations than the more popular genres. It is theorized that this is due to large entities such as radios or music groups focusing more on the popular genres rather than the less popular ones. As a result, these large entities make up the bulk of the postings for those genres. On the other hand, the less popular genres do not have as strong a backing of radio stations or music groups and therefore more individual users must be responsible for producing postings for that genre, making the distinct percentage higher.

The uniqueness of locations also follows this theory, since the same users most likely post from the same locations.

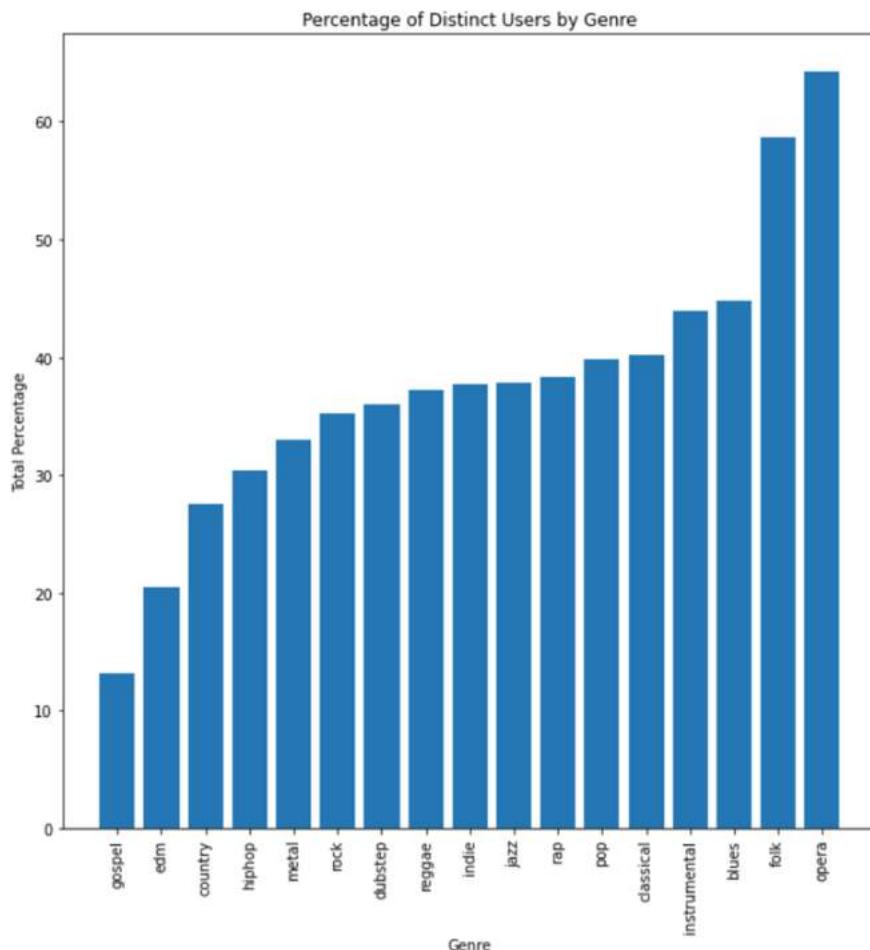


Fig. 2. Percentage of distinct users per genre, unsorted.

The source of the postings was also investigated since TAGS can pick up postings from platforms outside of Twitter (Fig. 4). Twitter obviously was the principal source, but other less-known platforms actually came in second which was quite interesting, beating out the third place Instagram by a decent margin. This helped move forward the idea that billboard charts most likely skip out on input from smaller sources such as these, and therefore gathering data from social media as is being done in this research is a good way to get a second angle.

Lastly, the counts of the friends and followers for each user per genre (Fig. 5) were analyzed. It was found that each genre's total friend and follower counts matched the popularity of the genre. This was expected, since more total postings were found for

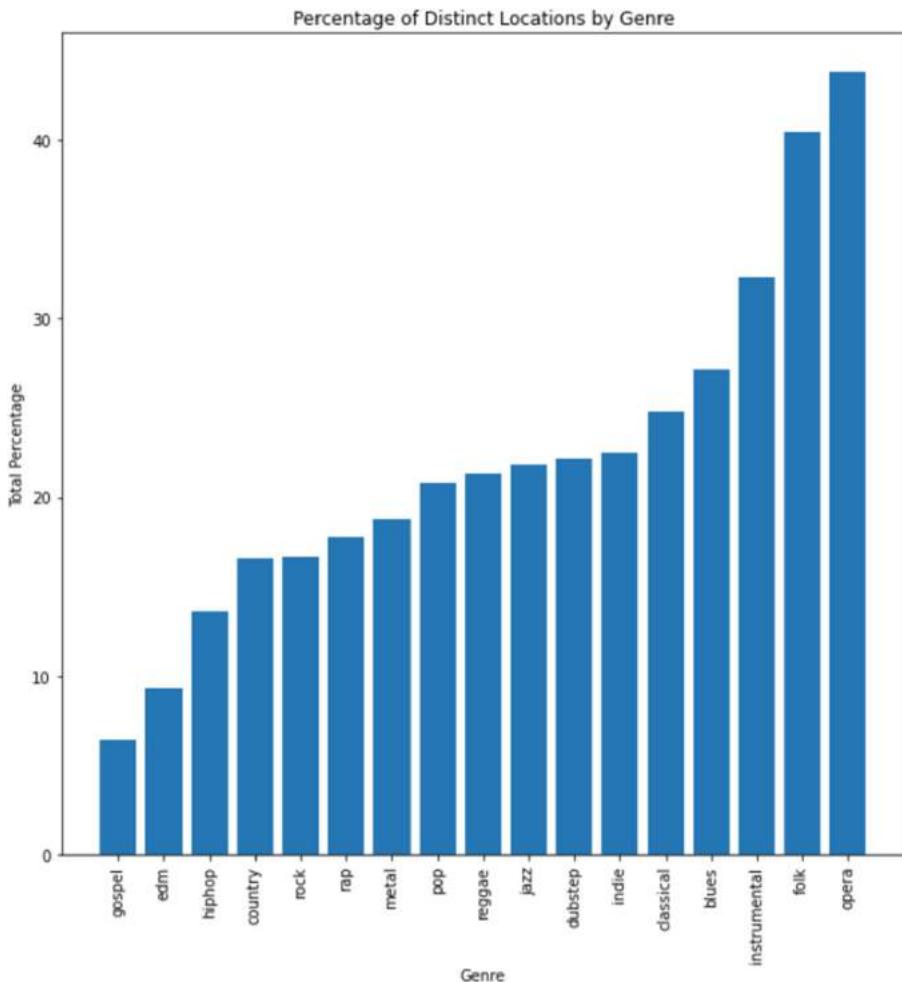


Fig. 3. Percentage of distinct locations per genre, unsorted.

more popular genres. However, it was also found that as a genre went up in popularity, its total follower count increased dramatically over its total friend count. For example, hip hop's total follower count was almost double that of its friend count, while folk's total follower count was only several thousand higher than the friend count. This could be due to the fact that users related to more popular genres are therefore more popular themselves with regard to strangers and get more followers than the less popular genre users as a result.

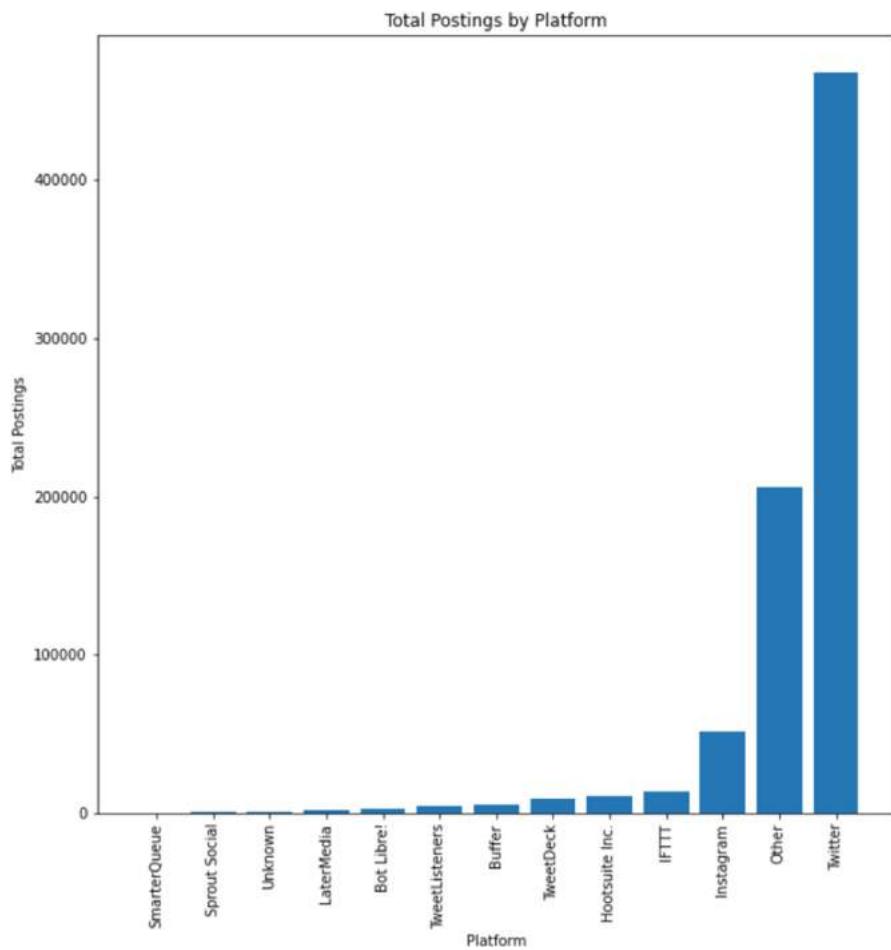


Fig. 4. Total postings per platform, sorted in ascending order. If no source was found it is listed as “unknown”, If it is not a major source it is listed under “other.”

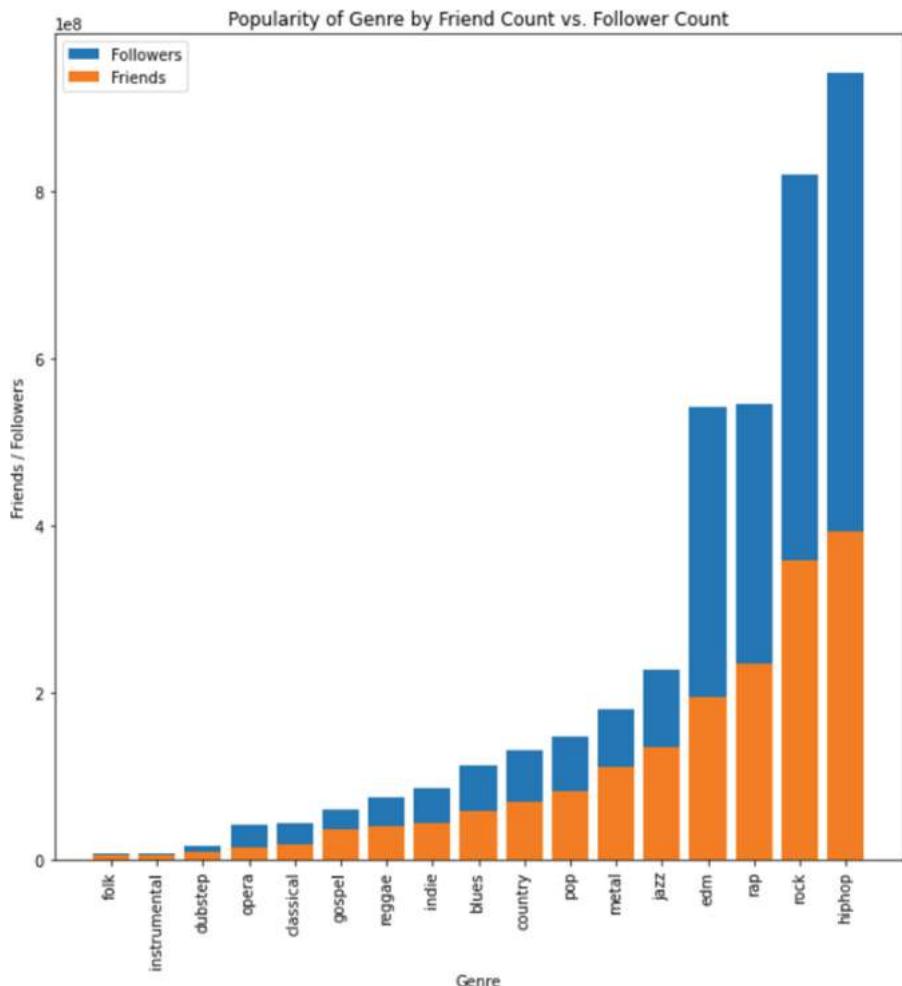


Fig. 5. Bar chart displaying total follower count (Blue) and total friend count (Orange) by genre.

6 Conclusion and Future Work

In conclusion, the results of the research show that hip hop is the most popular genre today, followed by rock and rap. They also show that many of the more classical or instrumental genres appear to be less popular than those focused more on beats and lyrics. For example, instrumental, classical, and opera are all near the bottom of the popularity chart. The attributes of the songs within these genres are thought of to be the reason for their low popularity compared to the more fast-paced, beat-centric genres such as rap, hip hop, and rock. The idea that billboard charts might be wrong was disproved as well. All in all, although billboards may skip over some smaller sources when making their rankings, they do match the results of the research performed in this paper. This not only helps validate billboard charts, but also adds to the idea that social media does in

fact follow trends and can be used to track what people are interested in their day-to-day lives.

For the future, it would be useful to find a more suitable method of gathering data more efficiently and effectively. Such a paper with the intention of using large-scale, big data should have data generated by a tool more fit for the task. Also, future work could include gathering more data from further back in the past, giving a clearer view of not only what genres are most popular today, but also what genres were more popular in the recent past. While this analysis would have to be done either with data only ranging back to the origin of Twitter (March 2006) or from billboards, it would be a good extended application of the initial motivation for the research at hand.

References

1. Andrews, T.: Billboard's charts used to be our barometer for music success. Are they meaningless in the streaming age? (2018). <https://www.washingtonpost.com/news/arts-and-entertainment/wp/2018/07/05/billboards-charts-used-to-be-our-barometer-for-music-success-are-they-meaningless-in-the-streaming-age/>
2. Beckwith, J.: The Evolution of Music Genre Popularity (2016). <https://thedataface.com/2016/09/culture/genre-lifecycles>
3. Hawksey, M.: TAGS (2010). <https://tags.hawksey.info/>
4. Sisario, B.: A Big Change Comes to Billboard's Album Chart: YouTube Streams (2019). <https://www.nytimes.com/2019/12/13/arts/music/billboard-youtube-charts.html>
5. Watson, A.: Music Genres Preferred by Consumers in the U.S. in 2018 (2019). <https://www.statista.com/statistics/442354/music-genres-preferred-consumers-usa/>
6. Zaharia, M., Chowdhury, M., Franklin, M.J., Shenker, S., Stoica, I.: Spark: cluster computing with working sets. HotCloud **10**(10-10), 95 (2010)



Result Prediction Using Data Mining

Hasan Sarwar¹, Dipannoy Das Gupta^{2(✉)}, Sanzida Mojib Luna³, Nusrat Jahan Suhi³, and Marzouka Tasnim³

¹ United International University, Dhaka, Bangladesh

² Edusoft Consultants Limited, Dhaka, Bangladesh

1405082.ddg@ugrad.cse.buet.ac.bd

³ Military Institute of Science and Technology, Dhaka, Bangladesh

Abstract. Data mining is being used in various fields to dig out important information; it can be very effective in the field of education as well for gaining important information from a large dataset that can be used to improve the educational environment. This paper is focused on an approach consisting of several well-known and widely used algorithms on training data set to predict students' grade for a particular course based on his/her previous results. Further analysis has been carried out considering several errors and accuracy factors of the resulted data in comparison with the actual data.

Keywords: Education · Result prediction

1 Introduction

Prediction of results of a student based on previous academic results is a fairly investigated topic in the research literature. Usually, students' performances or results are predicted from the results of previous semesters and other academic attributes using data mining techniques [1, 3–5, 7]. Traditionally results are still considered to be an indicator of student performance, especially, in the case of graduate students. So there are still enough opportunities to work deep in this field that will help students to get better grades in exams. Many factors have been identified that affect the performance of a student. The factors are not only limited to academic fields but also encompass socio-economic variables, backgrounds, and cultural parameters [3, 6].

Few works on education, mostly higher education, are found in the literature in various considerations. Brijesh and Saurabh [1] have analyzed students' performance (End Semester Marks) by fixing some variables related to student performance like previous semester mark, class test grade, assignment, etc. Suhirman, Zain, Haruna, Tutut [2] have presented a review on data mining. It may be used for supporting the academic decisions in educational field. The paper has discussed recent works on data mining in educational field and given outlines over future researches. Edin and Mirza [3] have done prediction on student performance not only with academic variables but also with socio-economic factors. They have worked with the impact of family, income, gender, high school results as well as current course attributes on a student's performance. Course grade was the

indicator of performance. Oyebade et al. [4] have used data mining for predicting the number of times a student will repeat a course. Neural network has been used as a data mining tool in this research. They have selected 30 attributes relating to the course itself, the teacher, and the particular student as predictor features. Behrouz, Deborah, Gerd, William [5] have presented a method to classify students in order to predict their final grades. The research has been executed depending on features extracted from logged data in an educational web-based system. The features used in this research are mostly connected with students' overall condition on academic performance including the number of corrected answers, time taken to answer a question, number of tries for homework, etc. Umesh and Pal [6] have shown a technique to find performer and underperformer of institutions using the Bayes Classification method. Here they have used caste, language, and class as attributes. Mueen, Zafar, Manzoor [7] have accomplished a study on the data set of two universities. This study has predicted students' academic performance based on general forum participation and academic attributes. Moreover, they have also shown a set of dominant predictor attributes in this performance prediction. All these works have tried to predict something regarding student performance or instructor performance.

Some university programs offer a fixed set of courses for a student for the next semester. Some other universities follow open credit type course offerings where a student is able to choose his or her desired courses from a list of offered courses. In the case of fixed selection, students generally have less options to choose his/her own course. In case of open credit, selection criteria for picking a course for the next semester vary among students. The interest for a particular topic or the intention to get easy marks may guide a student to select his or her next set of courses. Even in case of fixed setting, if a student could have been informed about the requirement of his effort to come up with a good result that would generate a positive contribution towards learning. Sometimes it gets too late to take proper preparation for a particular course and at the end of the semester; it is found that due to this course the result has turned unsatisfactory. No such studies are found that have analyzed the best sequence of suggestions. Another important criterion is that there are discipline-oriented courses that are tagged with one or more prerequisite courses. It means that a student can take a course if he or she has completed the required pre-requisite course(s). However, no such research is found where it shows that if there is any dependency of a course result on the already completed set of courses. Mostly, pre-requisite courses are set by using the experience of faculty members. Prerequisite courses are perceived as the foundation knowledge required to complete the main course. In this work, we have been able to show that student performance varies with an individual's achievement of grades of a set of courses in the earlier semesters. It means the grade of a course is affected by not only the pre-requisite course but also all the other courses that he or she has finished earlier on. To perform this task we have used data mining techniques and machine learning algorithms. This finding suggests that students will be better equipped in making a decision to select their courses for the next semester or put more effort on a particular course and be better able to come up with good results.

In this work, our goal is to make students enable to take more appropriate decisions with regard to emphasize on the right course during the start or throughout the semester. So we have tried to use established algorithms and found their effectiveness to predict

students' performance based on the previous courses which will in a way help students to achieve better result.

2 Classification Model

Classification is one of the most fundamental tasks of data mining. Classification is the process of predicting the class of some given data points. As an instance, classification model predicts any kind of category or class such as whether a fruit will be considered as an apple or a banana. Here the attributes of the fruits, namely, size, color, taste are used to predict fruit class. This simple concept of classifying an entity in a specific group can be extended to any other entities, which is the beauty of classification models. There are several classification algorithms or models in the field of machine learning that might be used for prediction.

2.1 Naïve Bayes

A Naive Bayes classifier is a probabilistic machine learning model. This classifier acts based on Bayes theorem. The assumption made here is that the predictors/features don't depend on each other. That is presence of one particular feature does not affect the other. Hence it is called naive.

2.2 J-48

Decision tree is another type of classification. There are two approaches of implementing a decision tree-based classifier. Univariate decision tree is one of them. Splitting is performed by using one attribute at internal nodes in this strategy. J48 algorithm is used to build such tree [8]. In this procedure, the first step is the construction of the tree. Second step is all about information gain. Third step consists of pruning.

2.3 K*

K* is a Heuristic Search Algorithm for Finding the k Shortest Paths. In the execution of K* algorithm, A* algorithm is used to search in graph G and Dijkstra to search in P(G) [9]. Here P(G) is a directed weighted graph formed from G. K* does not require the graph to be obviously available. Parts of the graph are generated when it is necessary. Another advantage is found due to the heuristic function. The function guides K* to perform better. These are the two advantages of K* over K shortest path [9].

2.4 Random Tree

Random tree is a set of large number of individual decision tree. All trees act like an ensemble. Random tree is also called random forest. Each individual tree in the random forest comes out with a class prediction. The tree which has most voted class becomes model's prediction [10].

3 Performance Measure Metrics

Some performance measure metrics are available to evaluate the performance of classification models.

3.1 Kappa Statistics

The kappa statistic is used to control only those instances that may have been correctly classified by chance. This can be calculated using both the observed (total) accuracy and the random accuracy.

3.2 Root Mean Square Error

In measuring the error of a model in predicting data, Root Mean Square Error (RMSE) is a standard approach. It actually indicates the deviation of predicted data from observed data. From the view of heuristic, RMSE can be illustrated as the difference between observed and predicted quantity. The concentration of data around the line of best fit can be deduced from RMSE [11].

3.3 Relative Absolute Error

Relative Absolute Error (RAE) is another procedure for measuring the performance of a classifier model. It is calculated with the following formula [12]:

$$RAE_i = \frac{\sum_{j=1}^n |P_j - T_j|}{\sum_{j=1}^n |T_j - T|} \quad (1)$$

Here P_j is the value, predicted by an individual program for j^{th} sample case out of n sample cases; the target value is expressed with T_j for sample case j ; and T is calculated by the formula [13]:

$$T = \frac{1}{n} \sum_{j=1}^n T_j \quad (2)$$

3.4 Root Relative Squared Error

The root relative squared error (RRSE) functions like Relative Absolute Error. More specifically, numerator is the total squared error and denominator is the total squared error of simple predictor. RRSE can be calculated with the following formula:

$$RRSE_i = \frac{\sqrt{\sum_{j=1}^n (P_j - T_j)^2}}{\sqrt{\sum_{j=1}^n (T_j - T)^2}} \quad (3)$$

Here P_j is the value, predicted by an individual program for j^{th} sample case out of n sample cases; the target value is expressed with T_j for sample case j ; and T is calculated by the formula [13]:

$$T = \frac{1}{n} \sum_{j=1}^n T_j \quad (4)$$

3.5 Info Gain

Information gain is an important quantity. It is found by calculating a value for a feature. More precisely, subtracting the entropy of the distribution after split from the entropy of the distribution before split, info gain is calculated. The largest information gain indicates smallest entropy.

3.6 Relief Attribute

Relief Attribute measures the utility of an attribute. For this purpose, repeated sampling of an instance is needed. Moreover value consideration of the given attribute for the nearest instance is required [14]. Both discrete and continuous class data can be evaluated with it [15].

3.7 False Positive and False Negative

False positive can be defined as receiving a positive result for an experiment, while negative result is expected. It's also being called as a false alarm or false positive error [16]. From the viewpoint of classification model, a false positive is a result where the model predicts the positive class incorrectly. And for a false negative outcome, classifier inaccurately predicts negative label or class.

3.8 Confusion Matrix

A confusion matrix is a table that is very useful to demonstrate the performance of a classification model. This special kind of prediction table is displayed in two dimensions. They are actual and predicted. With them, identical sets of “classes” also exist in both dimensions [17].

4 Data Collection

In this work, sample data has been collected from a sample university. As we have prioritized the courses, it was essential to conduct the collection process over the courses of a certain program. Moreover, consideration has been taken for multiple batches of students to keep the data size large enough for the convenience of patterns. However, there were a good amount of pitfalls like outliers, missing values for dealing with this large number of datasets.

4.1 Data Migration

For starting data preprocessing, we had to import the data set from “xls” sheet to the database server. For this purpose, we collected data from our sample university in “xls” format. Then the data has been migrated into the database server using the import feature of the database server.

4.2 Data Aggregation

After migration, we performed data aggregation. As the raw data was in the different sheets in xls file, they were imported into different tables. For example, student-course mapping, student-semester mapping, student-program mapping were contained in separate tables. All the required information for our training was gathered in a single table. Here we have used the typical database aggregate functions to accomplish this task. Moreover, the concept of PIVOT has also come in handy during the execution of row-column interchange.

4.3 Data Cleaning

After performing aggregation, we found lots of null grades against the courses which were necessary to remove for ensuring better performance of our classifiers. A student found with any null grade has been removed from the dataset. Moreover, there were many instances where a student took a course multiple times for improvement. Here we have applied a searching algorithm for finding out the best grade. After that, the remaining entries for that course have been deleted.

4.4 Outlier Detection and Replacement

A few numbers of outliers have been found in the dataset. For instance, some students were absent for a particular course while some students had drop course issues. We have tried to detect such anomalies in the dataset. As one of our main objectives was to keep the data size large, our task was not limited to outlier detection. Moreover, we had replaced those outliers with their actual grade secured in the subsequent semesters if it was available.

4.5 Attribute Selection

As we have stated earlier, we are analyzing previous semester course works, our selected attributes were courses. We have processed the raw data of the past three semesters to turn it into a structured one. In this process, a total of 16 courses of the previous three semesters have been selected to make the prediction of fourth-semester courses' grade and so the prediction of fourth-semester result.

4.6 Export Data

After the completion of all data preprocessing tasks in the database server, we got our structured, clean, and expected data set to feed into the classifiers. Here for the purpose of grade prediction, we have collected five data sets. Each data set consists of 18 variables. Among them, one is Student ID. The remaining 17 variables are different courses. Among them 16 courses are training attributes that are from previous semesters. And one course is a predicted variable that is registered for the new semester. We will predict this new semester course grade. After collecting data, we divided them into two parts. We kept 90% of them for training purposes and 10% of them for testing and analyzing

the accuracy. We have used J48, K-Star, Naive Bayes, and Random Tree for training and testing purposes. After conducting the training, we have applied them to the test data set and found out the predicted results from each algorithm. The predicted results were the Grade of each course of the new semester.

4.7 Input Parameter

For training phase, we have grades of sixteen different courses, student ID and the grade of the course we need as result of prediction.

Table 1. Sample of input parameters for training phase

Student ID	C1	C2	C3	C4	C16	C27
S1	A	B	A-	A+	B+	A+
S2	B-	A+	C	A	C	C
S3	A-	B-	A+	C	C+	B+
S4	A	A+	B	C	B-	C+

In Table 1, we can see a small sample of training dataset that consists of eighteen different attributes. Those are Student ID, grades of 16 different courses and the grades of course that we want to predict. Here last course C27 is the predicted course. Dotted portion indicates more courses. We have used other four different data sets for predicting grades of other four courses, with same attributes. The Student ID is consistent for all the data sets.

Table 2. Sample of matched instances

Student ID	Actual grade	Naïve Bayes
S1	A-	A
S2	C+	C

4.8 Output Parameter

After the training phase, testing has been done on 10% data. Then we have found predicted grades of each course using J48, Naive Bayes, K-Star and Random Tree algorithms. In Fig. 1, we can see a sample output format and it is the prediction of grade using one of the algorithms. For this sample, it is J48. As we had data set for five different courses, we gathered predicted results of those five courses for each algorithm.

After we got predicted grades for all the five courses we were aiming for, we calculated individual student's grade using a simple python program. For this phase, we used Student ID and the predicted grades of the courses as input.

After calculating grade, we have compared it with actual grade for finding out the percentage of matched instances.

In the Table 2, we can see three attributes student ID, their actual Grade and the Grade that was calculated from the predicted courses' Grade using Naive Bayes algorithm.

inst#	actual	predicted	error	prediction
1	2:'C	' 3:B+	+	1
2	4:'C+	' 3:B+	+	0.5
3	6:'A	' 9:D	+	1
4	5:'B	' 4:C+	+	0.5
5	5:'B	' 9:D	+	1
6	1:'A+	' 4:C+	+	1
7	1:'A+	' 1:A+		0.359
8	9:'D	' 7:C-	+	1
9	6:'A	' 1:A+	+	0.341
10	8:'A-	' 1:A+	+	0.341
11	5:'B	' 5:B		0.4
12	9:'D	' 2:C	+	0.333
13	1:'A+	' 1:A+		0.632
14	4:'C+	' 3:B+	+	0.5
15	3:'B+	' 6:A	+	1
16	1:'A+	' 8:A-	+	0.429
17	4:'C+	' 7:C-	+	0.4

Fig. 1. Sample of output parameters after prediction of individual course grade

5 Result and Analysis

In this study, we have worked with the data taken from four semesters. We have taken the courses of the fourth semester as our predicted attributes. There are 11 possible class labels both for the predicted and predictor attributes. All the labels are composed of different existing grades namely, A+, A, A–, B+, B, B–, C+, C, C–, D, F. We have to repeat the training and testing process for five times due to having five courses in the fourth semester. After each process, the individual grade of a particular course of the fourth semester is predicted. Finally, from predicted grades of five courses, we have calculated semester grade for the individual student and compared with actual grade.

The whole process is executed under four classification models which we have stated earlier. WEKA 4.8, [18] has been used as our testbed. Cross-validation of 10 folds has been used while training the data set. Here 10 folds means each time we have taken 10 instances from the data set and applied algorithms into them for training and testing the dataset. Same cross-validation has been applied for each of the classification algorithms.

Our preliminary goal was to compare the predicted grade with the actual grade of the fourth semester. As we completed the processes for all algorithms, we did the comparison. Here we have set the definition of success in two different scales. One is 100% matching or matching without error. Another one is matching with 10% error. 10% error means actual Grade is A–, but predicted Grade is A or B+. In the order of Grade,

A- comes after B+ and before A. Both grades are 1 unit distant from A-. This 1 unit far prediction has been considered as matching with 10% error.

From Tables 3, 4, 5 and 6, we can see that if we use Naive Bayes and Random tree algorithm in 0% error condition, it can detect 25% of students' grade accurately, where K-star and J48 have lower success rate here. But when we do the same procedure with considering 10% error then the success rate increases. Here Naive Bayes and J48 algorithm shows 61% and 53% success rate which are much better than the previous result. Better result with the increase of error rate is quite expected. From the above discussion, we can conclude that Naive Bayes is showing overall better performance than the remaining three.

Now we will analyze our above results with performance evaluation metrics.

Table 3. GPA comparison result with Naïve Bayes

Error rate	Total instance	Matched instance	Success rate
0%	54	14	25.92%
10%	54	33	61.12%

Table 4. GPA comparison result with K-Star

Error rate	Total instance	Matched instance	Success rate
0%	54	9	16.67%
10%	54	25	46.1%

Table 5. GPA comparison result with J48

Error rate	Total instance	Matched instance	Success rate
0%	54	10	18.51%
10%	54	29	53.70%

5.1 0% Error

With the 0% error, we have tested a total of 54 students' grades. Of these 54 students, Naïve Bayes could match with 14 students' grades. J48 could match with 10 students' grades. K Star could match with 9 students' grades. And Random Tree could match with 13 students' grades.

Table 6. GPA comparison result with Random Tree

Error rate	Total instance	Matched instance	Success rate
0%	54	13	24.07%
10%	54	26	48.4%

From Table 7, it is observed that Naïve Bayes is showing better performance from the perspective of 0% error. For Naïve Bayes three measurements of error (RMSE, RAE, RRSE) show less amount of error than the other three. Second to Naïve Bayes, Random Tree performs better. J48 and K* are ranked third and fourth respectively. Tables 8, 9 also reveal the best performance of Naïve Bayes. From the viewpoint of accuracy, precision, and recall Naïve Bayes dominates the other three. So Naïve Bayes is the best suit for this condition followed by Random Tree, J48, and K*.

Table 7. Performance measures of classifiers

Model	Kappa statistic	Root mean squared error	Relative absolute error	Root relative squared error
N. Bayes	0.75	5.44	40	74.07%
J48	0.83	5.99	44	81.48%
K*	0.81	6.124	45	83.33%
Rand. Tree	0.77	5.58	41	75.93%

Table 8. Performance measures of classifiers

Model	False positive FP	False negative FN	True positive TP	True negative TN	P = (TP + FP)	N = (TN + FN)
N. Bayes	40	14	14	40	54	54
J48	44	10	10	40	54	54
K*	45	9	9	45	54	54
Rand. Tree	41	13	13	41	54	54

Table 9. Performance measures of classifiers

Model	Accuracy $\frac{TP+TN}{P+N}$	Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{P}$	Specificity $\frac{TN}{N}$
N. Bayes	0.259	0.26	0.259	0.741
J48	0.185	0.19	0.185	0.814
K*	0.167	0.17	0.167	0.833
Rand. Tree	0.241	0.24	0.241	0.759

5.2 10% Error

With the 10% error, we have tested the same total of previously taken 54 students' grades. Among 54, Naïve Bayes could match with 33 students' grades. J48 could match with 29 students' grades. K Star could match with 25 students' grades. Random Tree could match with 26 students' grades.

From Table 10, it is observed that Naïve Bayes also performs better in the 10% error condition. For Naïve Bayes three measurements of error (RMSE, RAE, RRSE) show less amount of error than the other three. Second to Naïve Bayes, J48 performs better. Random Tree and K* are ranked third and fourth respectively. There is a swap in performance between Random Tree and J48 compared to the previous scenario of 0% error. However, Random Forest will stay ahead considering average error rates and success rate.

Table 10. Performance measures of classifiers

Model	Kappa statistic	Root mean squared error	Relative absolute error	Root relative squared error
N. Bayes	0.396	2.85	21	38.89%
J48	0.47	3.4	25	46.3%
K*	0.56	3.9	29	53.7%
Rand. Tree	0.53	3.8	28	51.85%

From Tables 11 and 12, it is also evident that Naïve Bayes is the best among the four models. From the view of accuracy, precision and recall, Naïve Bayes outperforms rest of the three classifiers. So Naïve Bayes can be chosen convincingly for this condition too. Then J48, Random Forest, K* will come respectively.

Though we have finally predicted the semester grade, it's not predicted in a direct manner. After predicting individual course grade, then the semester grade has been calculated. So under the abstraction of the semester grade, we have actually unfolded the new semester courses' grades. Our research goal is defined for two systems as we stated earlier. For the open credit system, our research can provide a set of courses for a

Table 11. Performance measures of classifiers

Model	False Positive FP	False Negative FN	True Positive TP	True Negative TN	P = (TP + FP)	N = (TN + FN)
N. Bayes	21	33	33	21	66	42
J48	25	29	29	25	58	50
K*	29	25	25	29	50	58
Rand. Tree	28	26	26	28	52	56

Table 12. Performance measures of classifiers

Model	Accuracy $\frac{TP+TN}{P+N}$	Precision $\frac{TP}{TP+FP}$	Recall $\frac{TP}{P}$	Specificity $\frac{TN}{N}$
N. Bayes	0.61	0.611	0.5	0.5
J48	0.53	0.537	0.5	0.5
K*	0.46	0.462	0.5	0.5
Rand. Tree	0.48	0.481	0.5	0.5

student which will be the best suit for him for the new semester based on the performance in his previous semester course works. If it is fixed credit system, our research will present a clear concept to the individual student regarding the preparation of the courses. And here is also the main factors are previous semester courses.

6 Conclusion

We have predicted the performance of students based on their previous academic records. Here a new idea has been tested and found to be meaningful as long as student result is important. We have shown that even there are little apparent dependencies of the result of one course on the previously completed courses; it is possible to predict the grade of the new course on the basis of the results of all the grades of previously taken courses. This result will help to guide a student to decide and dedicate his effort on a particular course and make him able to learn better. Our next aim is to train a large number of data in order to make the prediction more accurate along with considering some more features. There are scopes to consider other attributes to predict student grades. Moreover, the percentage of contribution of each predictor course on the outcome will also be determined in our future work.

References

1. Baradwaj, B.K., Pal, S.: Mining educational data to analyze students' performance. arXiv preprint [arXiv:1201.3417](https://arxiv.org/abs/1201.3417) (2012)
2. Zain, J.M., Herawan, T.: Data mining for education decision support: a review. Int. J. Emerg. Technol. Learn. **9**(6), 4–19 (2014)
3. Osmanbegovic, E., Suljic, M.: Data mining approach for predicting student performance. Econ. Rev.: J. Econ. Bus. **10**(1), 3–12 (2012)
4. Oyedotun, O.K., Tackie, S.N., Olaniyi, E.O., Khashman, A.: Data mining of students' performance: Turkish students as a case study. Int. J. Intell. Syst. Appl. **7**(9), 20 (2015)
5. Minaei-Bidgoli, B., Kashy, D.A., Kortemeyer, G., Punch, W.F.: Predicting student performance: an application of data mining methods with an educational web-based system. In: 33rd Annual Frontiers in Education, 2003, FIE 2003, vol. 1, pp. T2A–13. IEEE (2003)
6. Pandey, U.K., Pal, S.: Data Mining: A prediction of performer or underperformer using classification. arXiv preprint [arXiv:1104.4163](https://arxiv.org/abs/1104.4163) (2011)
7. Mueen, A., Zafar, B., Manzoor, U.: Modeling and predicting students' academic performance using data mining techniques. Int. J. Mod. Educ. Comput. Sci. **8**(11), 36 (2016)
8. Bhargava, N., Sharma, G., Bhargava, R., Mathuria, M.: Decision tree analysis on j48 algorithm for data mining. Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng. **3**(6), 1114–1119 (2013)
9. Aljazzar, H., Leue, S.: K*: a heuristic search algorithm for finding the k shortest paths. Artif. Intell. **175**(18), 2129–2154 (2011)
10. Yiu, T.: Towards data science-understanding random forest (2019). <https://towardsdatascience.com/understanding-random-forest-58381e0602d2>
11. Glen, S.: “RMSE: Root Mean Square Error” From StatisticsHowTo.com: Elementary Statistics for the rest of us! <https://www.statisticshowto.com/probability-and-statistics/regression-analysis/rmse-root-mean-square-error/>
12. Glen, S.: “Relative Absolute Error” From StatisticsHowTo.com: Elementary Statistics for the rest of us! <https://www.statisticshowto.com/relative-absolute-error/>
13. “Analyzing genexprotols models statistically,” <https://www.gepsoft.com/gxpt4kb/Chapter10/Section2/SS15.html>
14. Kira, K., Rendell, L.A.: A practical approach to feature selection. In: Sleeman, D.H., Edwards, P. (eds.) Ninth International Workshop on Machine Learning, pp. 249–256. Morgan Kaufmann (1992)
15. Kononenko, I.: Estimating attributes: analysis and extensions of relief. In: Bergadano, F., Raedt, L.D. (eds.) European Conference on Machine Learning, pp. 171–182. Springer (1994)
16. Hirai, M., Kawai, S.: False positive and negative results in allen test. J. Cardiovasc. Surg. **21**(3), 353–360 (1980)
17. Lewis, H., Brown, M.: A generalized confusion matrix for assessing area estimates from remotely sensed data. Int. J. Remote Sens. **22**(16), 3223–3235 (2001)
18. “WEKA The workbench for machine learning”. <https://www.cs.waikato.ac.nz/ml/weka/>



A Systematic Literature Review on Big Data Extraction, Transformation and Loading (ETL)

Joshua C. Nwokeji^(✉) and Richard Matovu

Gannon University, Erie, PA 16541, USA
{nwokeji001,matovu001}@gannon.edu

Abstract. Data analytics plays a vital role in contemporary organizations, through analytics, organizations are able to derive knowledge and intelligence from data to support strategic decisions. An important step in data analytics is data integration, during which historic data is gathered from various sources and integrated into a centralized repository called data warehouse. Although there are various approaches for data integration, Extract Transform and Load (ETL) has become one of the most efficient and popular approach. Over the decades, ETL has been applied to a wide range of domains such as finance, health and telecom to mention but a few. As the popularity and use of ETL grow, it becomes important to analyze and identify the trends in the research and practice of ETL. In this paper, we perform a systematic literature review to identify and analyze: (1) Approaches used to implement existing ETL solutions (2) Quality attributes to be considered while adopting any ETL approach. (3) The depth of coverage in ETL research and practice with regards to the application domains, frequency publications and geographical locations of papers. (4) The prevailing challenges in developing ETL solutions. Furthermore, we discuss the implications of our findings to ETL researchers and practitioners.

Keywords: ETL · Big data · Big data ETL · Systematic literature review

1 Introduction

Data analytics offers myriad advantages to modern-day organizations, for example, organizations are able to derive knowledge and intelligence to make strategic decisions through data analytics [5, 17]. In addition to decision support and other numerous advantages reported in literature [17], data analytics provides an avenue for organizations to make accurate forecast about sales, revenue, customer growth and other areas [5]. The data used in analytics is generated from various operational sources including e-commerce systems, Internet of Things (IoT) devices and social media [2]. In recent times, the amount of data has grown out of proportion in terms of size and volume, and thus, can no longer be

analyzed using traditional database management systems and analytics tools; *Big Data* is a term used to represent this type of data [3, 4, 9]. Big Data is characterized by 5 Vs namely: *Volume*, *Velocity*, *Variety*, *Veracity* and *Value*. (i) Volume: The huge amount of data generated from various transactional sources e.g., social media, satellite images etc. (ii) Velocity: The high speed at which these data are generated. (iii) Variety: The various formats in which data are generated. These can be structured, unstructured or combination of both. (iv) Veracity: The extent to which these data are trust worthy, truthful, and factual. (v) Value: The extent to which data provides intelligence to support business decisions [4, 9, 18].

The process of big data analytics or deriving knowledge and intelligence from data starts with data integration, an important activity during which data is gathered from a wide range of operational data sources such as e-commerce system and social media [6]. While there are various approaches for data integration, such as mediated query systems and federated databases [19], Extract Transform Load (ETL) has emerged as one of the most efficient and popular approach [4]. As the name implies, ETL data integration approach involves three phases i.e., extraction, transformation and loading of data [4]. During the extraction phase, data is extracted from multiple operational databases. This is followed by the transformation phase in which transformation rules and techniques (e.g., normalization and sorting) are defined and used to transform the extracted data. In the last phase, the transformed data is transferred or loaded into the data warehouse and made ready for analysis [4].

Due to the vital role ETL plays in data integration, business intelligence and data warehousing, it has received a considerable attention from scholars in recent times. This is easily seen by the increasingly number of studies and publications available in literature [1, 4, 6]. However, the majority of available studies in ETL focus on design and implementation of frameworks for optimizing ETL workflow and processes. For instance, Bensal [4] and Deb Nath *et al.* [6] proposed a semantic framework for optimizing and making ETL more efficient. Likewise, a model driven framework was proposed by Akkaoui [1] using business process modeling and notation (BPMN). This framework was aimed to support the implementation of ETL workflow during changes in organization's data as well as make ETL processes executable [1].

Currently, to the best of our knowledge, it is hard to find studies that clearly articulate available research in ETL with the objective of identifying and analyzing their contributions in terms of the methods and challenges faced in ETL. The aim of our study is to synthesize available publications in ETL and provide important insights into the current implementation approaches to ETL. In addition, our research puts in perspective, the existing challenges in ETL research and practice, thus helping upcoming researchers in ETL to focus on relevant problems. More so, we aim to categorize the quality attributes to be considered when selecting ETL solutions and presents current depth of ETL coverage in diverse areas e.g., application domain and geographical locations. In order to achieve the aim of our study, we use the Systematic Literature Review (SLR)

Methodology. The SLR performed here is an extension of preliminary work published in [11]. The current work significantly builds upon [11] as follows: (1) We broaden our research scope by addition of three (3) review questions to provide a more rigorous evaluation of ETL research in literature, (2) We discuss, in great details, the implications of our findings in relation to the various approaches, quality attributes, prevailing challenges and current trends of ETL solutions.

The remainder of this paper is organized as follows: In Sect. 2 we discuss how we applied the principles of systematic literature review to conduct this research. This is followed by Sect. 3, where we present the results derived from the primary studies. Subsequently, these results are discussed in Sect. 4, and finally, we conclude our research in Sect. 5.



Fig. 1. Systematic literature review process

2 Methodology

Our aim in conducting this study is to provide a critical perspective and useful insights on the current state of research in ETL, especially with regards to challenges, implementation approaches, quality attributes and the depth of coverage. This aim can be better achieved by conducting a systematic literature review (SLR). Following the description of SLR methodology provided by Kitchenham and Charters [10], we perform our SLR in six steps, these include define review objectives, specify review questions, develop search strategy, specify selection criteria, extract data, present results [10], see Fig. 1. These six steps are extracted from the three main phases (planning, conducting and reporting the review) of conducting SLR reported by Kitchenham and Charters, which also include the establishment of review protocol [10]. In the sections below, we explain the activities we performed in each step.

1. Define Review Objectives: The objectives of our systematic literature review relates to the overall aim of our study discussed in the preceding sections. These objectives include to articulate: *(i)* the main approaches or techniques used by scholars and practitioners to implement ETL workflows. *(ii)* ETL quality attributes. *(iii)* Prevailing challenges in ETL research. *(iv)* Current depth of coverage in ETL research, with focus on application domains, geographical locations and frequency of publications per year.

2. Specify Review Questions: We identified four (4) review questions that closely align with the review objectives defined above. These questions are as follows:

RQ1: What are the techniques or approaches used for implementing ETL workflows or processes?

RQ2: What are the quality attributes to be considered when selecting ETL solutions?

RQ3: What are the prevailing challenges in developing ETL solutions?

RQ4: What is the current depth of coverage in ETL research with respect to the application domain, geographical locations and frequency of papers published?

3. Develop Search Strategy: As recommended by Kitchenham and Charters [10], developing search strategy includes three activities namely, select data sources, identify search keywords, and conduct the search. In order to select our data sources, we considered databases that publish articles in computer and information science. These include IEEE Xplore, Science Direct, Google Scholar, ACM Digital Library and ProQuest. The next activity is to identify keywords to use for our search. In order to identify keywords, we first conducted pilot searches using initial keywords identified from our review questions and objectives. These pilot searches helped us to identify more keywords e.g., synonyms that relates to the initial keywords. These concepts and their corresponding keywords are shown in Table 1. Finally, we conducted the search by combining each keywords in concept [A], see Table 1, with the keywords in concepts [B], and [C] using boolean operators (and/or).

Table 1. SLR keywords

Main concept	Corresponding keywords
[A] Extract, Transform, Load	[A1] Extract*, Transform*, Load*; [A2] ETL
[B] Approach	[B1] Framework; [B2] Models; [B3] Tools; [B4] Technology; [B5] Software
[C] Quality	[C1] Quality Attributes; [C2] Quality Measures; [C3] Quality Features; [C4] Quality Characteristics; [C5] QoX

4. **Specify Selection Criteria:** Search results usually include more papers than are necessary. Hence, specifying criteria for selecting a paper for review is an important aspect of SLR. For our study, we identified, discussed and agreed on the following selection criteria: year, relevance, language, accessibility, quality and rigor of papers. In Table 2, we provide descriptions of these criteria.

Table 2. Selection criteria

Criteria	Include	Exclude
Year	Papers must be published between 2006 and 2016	Papers published before 2006
Relevance	Papers whose title and abstracts are relevant to ETL and its related concepts, as well as the review objectives and questions	Papers that do not relate to the ETL and other key concepts of this research
Language	Papers in English Language	Papers in other languages
Accessibility	Papers whose full text are accessible and available to the authors at the time of search	Papers whose full text are not accessible and available at the time of search
Quality	Peer reviewed papers in journals, conferences, and workshops	Non-peer reviewed papers. Also keynotes and presentations
Rigor	Papers that demonstrate rigor through the use of appropriate scientific research and validation methods	Papers that do not use appropriate scientific research and validation methods

5. **Extract Data:** In Fig. 2, we summarize the steps and activities involved in our data extraction process. The first step include searching the data sources with the selected keywords shown in Table 1, this returns a total of 865 papers. These papers were extracted into *Mendeley*-a bibliography management software. In Step 2, we removed duplicate papers *i.e.*, papers that appear in two or more data sources; for instance, most papers in ACM Digital Library and IEEE Xplore were also found in Google Scholar. This is followed by Step 3, where we excluded papers that do not meet the year criterion described in Table 2. Afterwards, we applied the relevance criteria in Steps 4 to exclude papers whose titles and abstracts are not related to the main concepts of the review. In Step 5, we excluded papers based on the language and accessibility criteria described in Table 2. Finally, we selected a total of 97 papers as primary studies, after we applied the quality and rigor criteria respectively in Step 6.

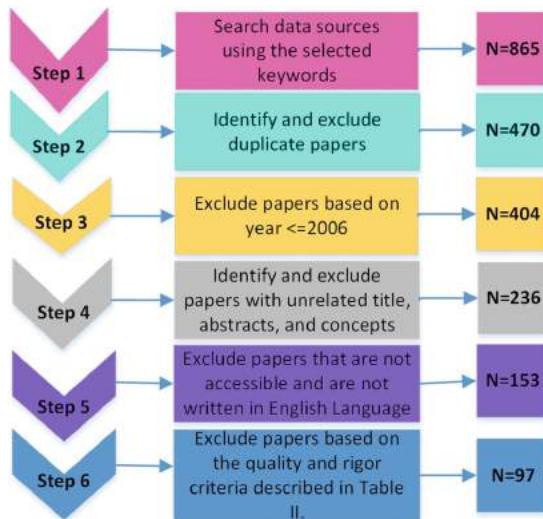


Fig. 2. Data extraction process

3 Result and Answers to Review Questions

In this section, we present the results of our SLR and use these to answer the research questions. The papers selected as primary studies, (*i.e.*, *S1* to *S97*) including their year of publication are shown in Table 3.

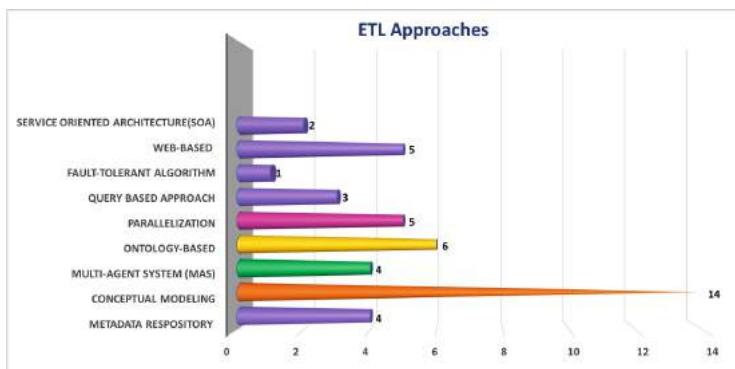


Fig. 3. Approaches used to implement ETL solutions

Table 3. Papers selected for review (primary studies)

Paper ID, title and year	Paper ID, title and year	Paper ID, title and year
[S1] A domain ontology approach in the ETL process of data warehousing (2010)	[S34] Big-ETL: Extracting-Transforming-Loading Approach for Big Data (2015)	[S67] Representing interoperable provenance descriptions for ETL workflows (2012)
[S2] A framework for interoperable distributed ETL components based on SOA (2010)	[S35] Bringing business objects into extract-transform-load (ETL) technology (2008)	[S68] Research and Implementation of a Universal ETL Management Platform based on Telecom Industry (2009)
[S3] A BPMN-based design and maintenance framework for ETL processes (2013)	[S36] Cloud computing based ETL technique using Warehouse Intermediate Agents (2011)	[S69] Research of distributed ETL engine based on MAS and data partition (2011)
[S4] A content-driven ETL processes for open data (2015)	[S37] ColudETL: Scalable Dimensional ETL for Hive Categories and Subject Descriptors (2014)	[S70] Research on the Heterogeneous Data Integration Platform Based on Extended ETL and Data Federate (2011)
[S5] A framework for detecting unnecessary industrial data in ETL processes (2014)	[S38] Defining ETL workflows using BPMN and BPEL (2009)	[S71] Research on the stream ETL process (2014)
[S6] A Framework for User-Centered Declarative ETL Functionality-Based Design (2014)	[S39] Defining global schema for ETL of human resource performance appraisal system using REA ontology (2011)	[S72] Scheduling strategies for efficient ETL execution (2013)
[S7] A framework model study for ontology-driven ETL processes (2008)	[S40] Design and Realization of Bank Non-site Auditing ETL (2013)	[S73] Semi-streamed index join for near-real time execution of ETL transformations (2011) S73
[S8] A framework study of ETL processes optimization based on metadata repository (2010)	[S41] E-ETL: Framework for managing evolving ETL processes (2011)	[S74] Systematic ETL management - Experiences with High-level Operators (2013)
[S9] A model-driven framework for ETL process development (2011)	[S42] ETL Design Toward Social Network Opinion Analysis (2016)	[S75] stream Loader: An Event-Driven ETL System for the On-line Processing of Heterogeneous Sensor Data (2016)
[S10] A new tool for ETL process (2012)	[S43] ETL Techniques and Challenges in Agriculture Intelligence (2012)	[S76] SPOOL: a SPARQL-based ETL Framework for OLAP Over Linked Data (2015)
[S11] A novel agent-based parallel ETL system for massive data (2016)	[S44] ETL tool research and implementation based on drilling data warehouse (2010)	[S77] The Management of Conformed ETL Architecture (2015)
[S12] A PaaS based metadata-driven ETL framework (2011)	[S45] ETL workflow analysis and verification using backwards constraint propagation (2009)	[S78] The research and application of ETL tool in business intelligence project (2009)
[S13] A QoX model for ETL subsystems: Theoretical and industry perspectives (2013)	[S46] Fact - Centered ETL: A Proposal for Speeding Business Analytics up (2014)	[S79] The Research and Application of an ETL Model Based on Task (2009)
[S14] A Semantic approach towards CWM-based ETL processes (2008)	[S47] HotDataSpider, an ETL tool for supplementary data of biomedical journals (2009)	[S80] Towards a Programmable Semantic Extract-Transform-Load Framework for Semantic Data Warehouses (2015)
[S15] A Triggering and scheduling approach for ETL in a real-time data warehouse (2010)	[S48] Implementation of web-ETL transformation with pre-configured multi-source system connection and transformation mapping statistics report (2010)	[S81] Towards Generating ETL Processes for Incremental Loading (2008)
[S16] A Unified Model Driven Methodology for Data Warehouses and ETL Design (2011)	[S49] Integrating big data: A semantic extract-transform-load framework (2015)	[S82] Unified Views: An ETL Tool for RDF Data Management (2016)

(continued)

Table 3. (*continued*)

Paper ID, title and year	Paper ID, title and year	Paper ID, title and year
[S17] Agile ETL (2013)	[S50] Integrating Energy Data with ETL (2012)	[S83] Using ETL tools for developing a virtual data warehouse (2016)
[S18] An Approach for Designing, Modeling and Realizing ETL Processes Based on Unified Views Model (2011)	[S51] Integrating ETL processes from information requirements (2012)	[S84] Using Relational Algebra on the Specification of Real World ETL Processes (2015)
[S19] An approach to conceptual modelling of ETL processes (2014)	[S52] KANTARA: A Framework to Reduce ETL Cost and Complexity (2016)	[S85] Using reo on etl conceptual modelling: a first approach (2013)
[S20] An enhanced extract-transform-load system for migrating data in telecom billing (2008)	[S53] Lazy ETL in Action: ETL Technology Dates Scientific Data (2013)	[S86] Web based ETL component extended with loading and reporting facilitating a financial application tool (2010)
[S21] An ETL Framework Based on Data Reorganization for the Chinese Style Cross-tab (2011)	[S54] Leveraging business process models for ETL design (2010)	[S87] Towards a matrix based approach for analyzing the impact of change on ETL processes (2011)
[S22] An ETL Optimization Framework Using Partitioning and Parallelization (2015)	[S55] Macro-level Scheduling of ETL Workflows (2009)	[S88] Towards a low cost ETL system (2014)
[S23] An ETL services framework based on metadata (2010)	[S56] Managing ETL Processes (2008)	[S89] Significance of data integration and ETL in business intelligence framework for higher education (2016)
[S24] An integrated use of CWM and ontological modeling approaches towards ETL processes (2008)	[S57] MapReduce-based Dimensional ETL Made Easy (2012)	[90] Semantic ETL into i2b2 with Eureka! (2013)
[S25] An Intelligent ETL Grid-Based Solution to Enable Spatial Data Warehouse Deployment in Cyber Physical System Context (2015)	[S58] Modeling and Supporting ETL Processes via a Pattern-Oriented, Task-Reusable Framework (2014)	[S91] Semantic Analysis for an Advanced ETL framework (2009)
[S26] An intelligent ETL workflow framework based on data partition (2010)	[S59] Modeling Extraction Transformation Load Embedding Privacy Preservation using UML (2012)	[S92] Quality measures for ETL processes: from goals to implementation (2016)
[S27] An MAS-based and fault-tolerant distributed ETL workflow engine (2012)	[S60] Modelling ETL Conciliation Tasks Using Relational Algebra Operators (2014)	[S93] Pygrametl: A Powerful Programming Framework for Extract-transform-load Programmers (2009)
[S28] An Object-Oriented modeling and Implementation of Web based ETL process (2010)	[S61] Optimization of ETL Work Flow in Data Warehouse (2012)	[S94] Prototype of a Web ETL Tool (2014)
[S29] An Open Source ETL Tool-Medium and Small-scale Enterprise ETL (MaSSEETL) (2014)	[S62] Ontology-based conceptual design of ETL processes for both structured and semi-structured data (2007)	[S95] POIESIS: a Tool for Quality-aware ETL Process Redesign (2015)
[S30] An optimized ETL fault-tolerant algorithm in data warehouses (2013)	[S63] Novel approach in ETL (2013)	[S96] Optimizing ETL by a Two-level Data Staging Method (2016)
[S31] Application of ontology-based automatic ETL in marine data integration (2012)	[S64] Partitioning real-time etl workflows (2010)	[S97] On-demand ELT architecture for right-time BI, extending the vision (2013)
[S32] Automatic composition of ETL workflows from business intents (2013)	[S65] P-ETL: Parallel-ETL based on the MapReduce paradigm (2014)	
[S33] Automatic generation of ETL processes from conceptual models (2009)	[S66] QoX-driven ETL design: Reducing the cost of ETL consulting engagements (2009)	

3.1 RQ1: What Approaches are Currently Used to Implement ETL Solutions?

As shown in Fig. 3 below, we found nine (9) popular implementation approaches from selected papers (primary studies). These include approaches that are implemented using: (i) Service oriented architecture (SOA), as reported in *S2 and S40*; (ii) Web-based technologies, e.g., semantic web, these are reported in *S7, S28, S48, S86, and S94*; (iii) Fault-tolerant algorithm, which is reported in *S30*; (iv) Structured query languages (SQL), as reported in *S53, S60, and S84*; (v) Parallelization (parallel computing paradigm), e.g., Map Reduce, these are reported in *S22, S34, S37, S57, and S65*; (vi) Domain ontology, which are reported in *S1, S14, S24, S31, S39, S62*; (vii) Multi-agent system (MAS), as reported in *S11, S25, S27, S36*; (viii) Conceptual modeling e.g., unified modeling language (UML) and business process modeling (BPMN), as reported in *S3, S9, S16, S18, S19, S32, S33, S38, S46, S49, S54, S58, S59, S61*; and finally (ix) Meta-data repository, these are reported in *S8, S12, S44, S79*.

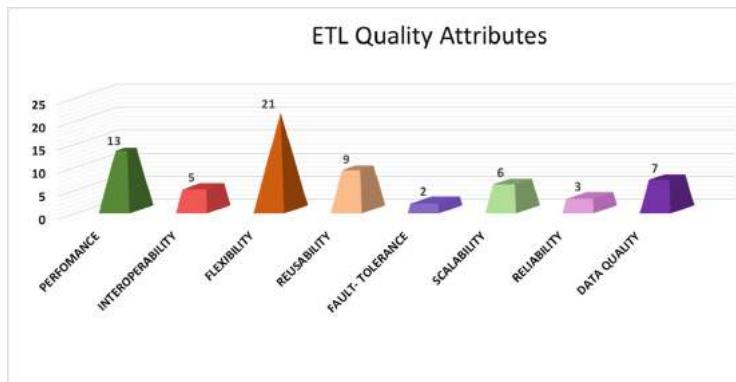


Fig. 4. Quality attributes to be considered

3.2 RQ2: What are the Quality Attributes to be Considered When Designing ETL Solutions?

Figure 4 above shows the 8 quality attributes identified from primary studies. They include (i) Performance, which is reported in *S5, S9, S11, S17, S27, S30, S34, S37, S48, S61, S64, S65, and S86*. Performance implies that the ETL process must be completed within a specified time window and also the functional mapping of data from source to target must be correct [14]; (ii) Interoperability, as reported in *S9, S14, S16, S39, and S56*, implies that ETL solutions should support the integration of data from heterogeneous operational data sources, irrespective of technological, syntactic, and semantic schemas [8,15]; (iii) Flexibility, this is reported in *S2, S7, S12, S23, S28, S38, S40, S44, S48, S50, S56, S58, S59, S64, S68, S70, S77, S79, S87, S95, and S97*, refers to the ability of an

ETL solution to accommodate changes in data integration requirements [5]; (iv) Reusability, as reported in *S2*, *S5*, *S16*, *S28*, *S33*, *S36*, *S52*, *S58*, *S69*, defines the ability of an ETL solution to be used in various application domains [16]; (v) Fault-tolerance i.e., the ability of ETL solution to function optimally in the presence of fault [16], is reported in *S30* and *S79*; (vi) Scalability, which is reported in *S2*, *S36*, *S53*, *S57*, *S65*, *S69*, defines the ability of ETL solution to be used for data of varying volumes and complexities [5]; (vii) Reliability, this is reported in *S14*, *S68*, *S79*, is the probability that ETL solution will not fail to perform its intended functions within the specified time [5]; (viii) Data quality, refers to the degree to which data produced by an ETL solution is fit for use [16], this is reported in *S2*, *S5*, *S16*, *S28*, *S33*, *S36*, *S52*.



Fig. 5. Challenges in ETL research and practice

3.3 *RQ3: What are the Prevailing Challenges in Developing ETL Solutions?*

Figure 5 shows a word cloud summarizing the challenges identified from the primary studies considered in our work. We identified 7 key challenges in developing ETL solutions. These include complexity, data heterogeneity, cost, time, maintenance, lack of automation and standardization issues. In Table 4, we describe each of these challenges and list the primary studies that report them.

3.4 RQ4: What is the Current Depth of Coverage in ETL Research with Respect to the Application Domain, Geographical Locations and Frequency of Papers Published?

As shown in Fig. 6, we identified 6 domains where ETL solutions are applied. These include data warehouse (reported in 63 papers); business intelligence (reported in 12 papers); Big data (reported in 4 papers); health (reported in 4 papers); telecommunications (reported in 2 papers) and finance (reported in 2

Table 4. Prevailing challenges in ETL

Challenges	Brief description	Papers
Complexity	Due to the exponential increase in volumes, data is becoming more complex. Hence, it is becoming more difficult to develop effective ETL solutions	S12, S18, S22, S25, S34, S50, S51, S52, S54, S58, S59, S61, S66, S67, S72, S74, S75, S79, S81, S85, S88, S90, S92, S94
Data heterogeneity	Data is generated from different sources and formats. Thus developing ETL solution for effective integration is difficult	S1, S4, S7, S14, S24, S25, S29, S31, S36, S39, S49, S62, S67, S84
Costly	Developing ETL solutions is a cost intensive process	S28, S36, S48, S52, S53, S60, S61, S66, S86
Time-Consuming	Significant amount of time is required to design, develop, implement, and execute ETL solutions	S5, S9, S17, S22, S28, S32, S48, S51, S52, S53, S60, S63, S66, S68, S72, S86, S88, S94
Maintenance	Difficult to maintain the already developed ETL solutions due changing data schemas and business requirements	S3, S18, S19
Lack of automation	Most ETL solutions still depends on manual process or requires human intervention; while others are partly automated	S6, S16, S32, S33, S45, S60
Standardization issues	There is no standardized approach to modeling ETL process and execution work flows	S42, S56, S66, S84

papers). In order to optimize space, we omitted the list of the primary studies for *RQ3*.

Figure 7 shows the number of papers published per geographical locations (continents). We identified 4 geographical locations that actively participate in ETL research and practice. These include Europe (38 papers), Africa (6 papers), North America (13 papers), and Asia (40 papers). Finally, we show the frequency of papers published per year in Fig. 8.

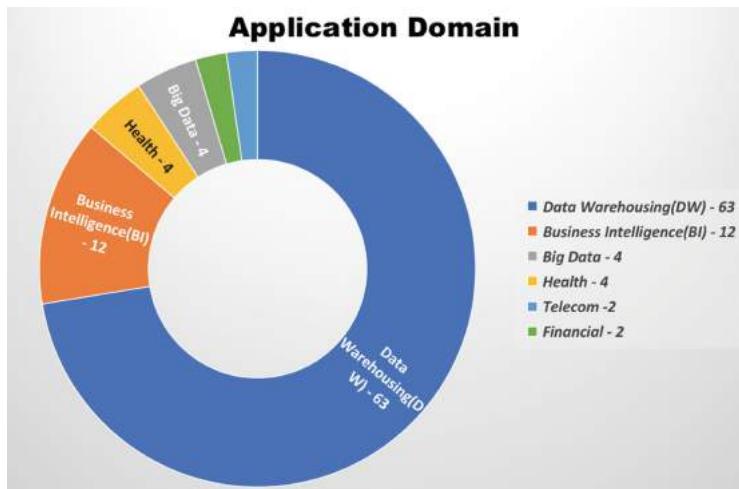


Fig. 6. ETL application domain

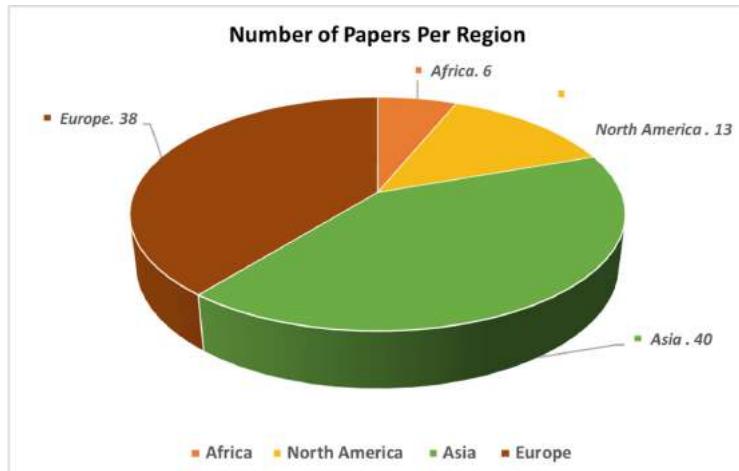


Fig. 7. Number of papers published per geographical location

4 Discussion of Results

4.1 ETL Implementation Techniques

One of the main objective of our study is to articulate the current implementation techniques of ETL workflow. Our systematic review shows that a variety of techniques are used by researchers and practitioners to develop ETL workflows. In Fig. 3, we use a graph to represent these techniques and the number of times each of them are used in our primary studies. We found that conceptual modeling

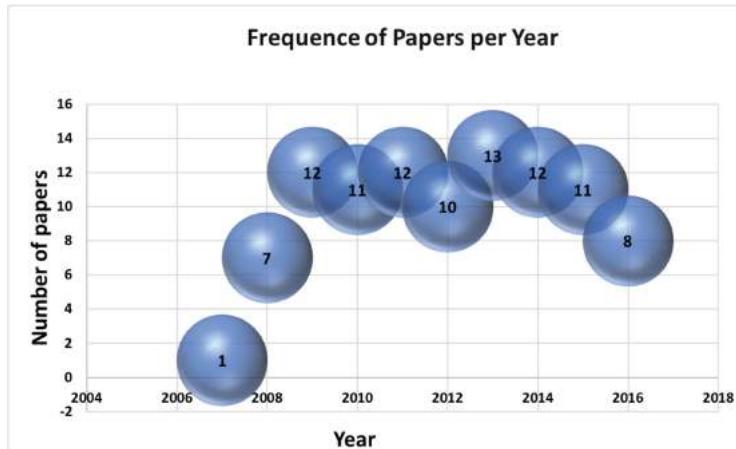


Fig. 8. Frequency of papers published per year

is widely used to design and implement ETL workflows, for instance, see the following studies in Table 3 [S3](#), [S9](#), [S16](#), [S18](#), [S19](#), [S32](#), [S33](#), [S38](#), [S46](#). Some of these conceptual modeling technique include the unified modeling languages (UML) e.g., see [S16](#),[59](#); model driven architecture e.g., see [S9](#),[S33](#), and business process modeling and notation (BPMN) e.g., see [S3](#), [38](#). A conceptual model is an explicit representation of the real world at various levels abstractions using concepts [12].

Conceptual models are beneficial since they facilitate the representation, analysis and easy understanding of complex ideas. Other advantages of conceptual modeling include support for automatic code generations [7]. Although the application of conceptual modeling to ETL workflows can be beneficial, the overly focus on this at the expense of other approaches calls for concerns. Particularly, it is not clear from literature how these conceptual models can be scaled to accommodate the ever growing data complexity, heterogeneity and volume. The suitability of conceptual model as ETL techniques in the future is unclear. Furthermore, existing approaches did not give due considerations to the dimensions of organizational changes, and how ETL solutions can support the ability to respond to these changes effectively, otherwise called organizational agility [13]. Agility is desirable capability to enterprises but challenging

Also concerning and surprising is the lack of new and emerging technologies such as machine learning, artificial intelligence and robotics as techniques for ETL implementation. It will be helpful for more research attention to be given on if and how these emerging technologies can be used to implement ETL workflows. For instance, how can artificial intelligence or machine learning be used to automate the data extraction from heterogeneous sources? Research should also be conducted to identify the advantages and disadvantages of using these emerging technologies in ETL implementation. Our expectation is that future

research endeavors in ETL workflows should focus on implementing techniques that leverage of on emerging technologies.

4.2 Quality Attributes

Among the quality attributes identified and presented in Fig. 4, fault-tolerance *i.e.*, the ability of ETL solution to function optimally in the presence of error [16] is the least. Similarly, fault-tolerant algorithm is also the least approach used to develop ETL solutions, see Fig. 3. It is not clear why existing research is under-emphasize the importance of fault-tolerance both as an approach to developing ETL solutions and a desirable quality attribute for ETL solution. Equally, reliability is another quality attribute that received less attention from existing studies, see Fig. 4. In this context, reliability refers to the probability that ETL solution will not fail to perform desired functions within a specified period of time [5].

The relative importance of fault-tolerance and reliability is obvious from their definition. Hence, one would expect these (fault-tolerance and reliability) to be among the top quality attributes to be considered in ETL solution. However this is not the case, and the reasons for this is not made obvious in existing publications. There is a possibility that fault-tolerance and reliability are not necessary quality attributes in ETL, there is also another possibility that these are erroneously under-emphasized in literature. It would therefore be useful and necessary to carry out further investigations that explains why fault-tolerance and reliability are less emphasized as ETL quality attributes.

4.3 Prevailing Challenges

Investment in resources and efforts have been made in research and practice to provide effective solutions to the challenges in big data integration using ETL. Yet, as shown in Table 4, existing studies show that some of these challenges are still prevailing. The most prevailing challenges identified from this SLR are data complexity and heterogeneity. Data complexity refers to the exponential growth in the volume of data generated by organizations; while data heterogeneity refers to the diverse transactional sources from which data is generated e.g., social media, satellite, smart phones, etc.

These two challenges are most likely to be exacerbated over time, since data growth and complexity will exponentially increase in direct proportion with time; moreover, newer technologies that generate data are developed at increasing rate [9]. Addressing these challenges would require the amplification of the concerted effort currently existing between researchers and practitioners. Such effort should focus on the development newer, innovative and efficient approaches to ETL solutions at a faster rate to match the rate at which data is growing in complexity as well as heterogeneity of sources. Considerations should also be given to other challenges such as developing automated approaches to ETL solutions, standardized modeling and cost effective approaches to ETL solutions.

4.4 Current Trends

As shown in Fig. 6, ETL is most applicable to data warehouse and business intelligence. But most studies do not specify the specific industries where ETL approaches are used to develop data warehousing and business intelligence solutions. This may explain why the application of ETL approaches to industries such as health, telecom, and finance is underrepresented, as shown in Fig. 6. Figure 7 shows a lesser participation in ETL research from Africa and North America. Effort should be made to increased ETL and big data related research in these areas to compensate for this poor participation. Lastly, as seen in Fig. 8, ETL research activities and publications are on steady decline from 2013 to 2016. Specific reasons for these are not clear and should be a focus of future research. However, it is possible that the advent of emerging computing technologies such as artificial intelligence, and machine learning has overshadowed ETL research and practice.

5 Conclusion and Future Work

Data extraction, transformation and loading (ETL) is a popular technique used in data integration i.e., gathering data from various operational sources into the data warehouse [6]. Due to its vital role in data integration, ETL is receiving an increasing attention in research and practice, yet fewer studies have synthesized existing studies in ETL to define future research direction that is robust and sustainable. Hence, the need for the systematic literature review presented in this paper. The scope of our systematic review as shown in our research questions in Sect. 2, is to articulate the current contributions in ETL with regards to the techniques used to implement ETL workflows, the quality attributes to consider when selecting ETL implementation techniques as well as the prevailing challenges in ETL research and practice.

In terms of primary studies, our systematic review is based on papers published between 2006 and 2016. Initially, we identified $N = 865$ papers, after applying our inclusion and exclusion criteria, we selected $n = 97$ papers as primary studies. Based on these papers we found that the current techniques for implementing ETL solutions overly focus on conceptual modeling, neglecting emerging and innovative approaches such as artificial intelligence and machine learning. (ii) Data complexity and heterogeneity are key challenges to ETL. Given that big data is most likely to grow exponentially (in complexity and heterogeneity) in direct proportion with time; these challenges will be chronically predominant over time. (iii) Much attention has not been given to some important ETL quality attributes such fault-tolerance and reliability.

These challenges provide motivations for more research on ETL trends and implementation techniques. Ideally, robust and sustainable ETL implementation techniques would be required to solve the problems of big data integration and management. In the future, we plan to extend the scope of our systematic review to include more research questions. Also, we plan to start developing solutions to the challenges reported in this paper. For instance, we will focus on

how emerging and innovative technologies such as robotics, artificial intelligence, machine learning can be applicable in ETL. Also, we are interested in on how data security can be strengthened in the ELT process.

References

1. El Akkaoui, Z., Zimànyi, E., Mazón, J.N.: A model-driven framework for ETL process development. In: Proceedings of the ACM (2011)
2. Aqlan, F., Nwokeji, J.C.: Applying product manufacturing techniques to teach data analytics in industrial engineering: a project based learning experience. In: 2018 IEEE Frontiers in Education Conference (FIE), pp. 1–7, October 2018
3. Aqlan, F., Nwokeji, J.C., Shamsan, A.: Teaching an introductory data analytics course using microsoft access® and excel®. In: 2020 IEEE Frontiers in Education Conference (FIE), pp. 1–10, October 2020
4. Bansal, S.K.: Towards a semantic extract-transform-load (ETL) framework for big data integration. In: 2014 IEEE International Congress on Big Data, pp. 522–529, June 2014
5. Dayal, U., Castellanos, M., Simitsis, A., Wilkinson, K.: Data integration flows for business intelligence. In: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, EDBT 2009, pp. 1–11. ACM, New York (2009)
6. Deb Nath, R.P., Hose, K., Pedersen, T.B.: Towards a programmable semantic extract-transform-load framework for semantic data warehouses. In: Proceedings of the ACM Eighteenth International Workshop on Data Warehousing and OLAP, DOLAP 2015, pp. 15–24. ACM, New York (2015)
7. El Akkaoui, Z., Zimanyi, E., Mazon Lopez, J.N., Trujillo Mondejar, J.C., et al.: A BPMN-based design and maintenance framework for ETL processes. *Int. J. Data Warehous. Min.* **9**, 46–72 (2013)
8. Freitas, A., Kampgen, B., Oliveira, J.G., ORiain, S., Curry, E.: Representing interoperable provenance descriptions for ETL workflows. In: Extended Semantic Web Conference, pp. 43–57. Springer (2012)
9. Gudivada, V.N., Baeza-Yates, R.A., Raghavan, V.V.: Big data: promises and problems. *IEEE Comput.* **48**(3), 20–23 (2015)
10. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering version 2.3. *Engineering* **45**(4ve), 1051 (2007)
11. Nwokeji, J.C., Aqlan, F., Olagunju, A.: Big data ETL implementation approaches: a systematic literature review (P) (2018)
12. Nwokeji, J.C., Aqlan, F., Barn, B., Clark, T., Kulkarni, V.: A modelling technique for enterprise agility. In: Proceedings of the 51st Hawaii International Conference on System Sciences (2018)
13. Nwokeji, J.C., Clark, T., Barn, B., Kulkarni, V.: A conceptual framework for enterprise agility. In: Proceedings of the 30th Annual ACM Symposium on Applied Computing, pp. 1242–1244. ACM (2015)
14. Simitsis, A., Wilkinson, K., Dayal, U., Castellanos, M.: Optimizing etl workflows for fault-tolerance. In: 2010 IEEE 26th International Conference on Data Engineering (ICDE 2010), pp. 385–396, March 2010
15. Teodoro, D.H., et al.: Interoperability driven integration of biomedical data sources. *Stud. Health Technol. Inf.* **169**, 185–9 (2011)

16. Theodorou, V., Abelló, A., Lehner, W.: Quality measures for ETL processes. In: International Conference on Data Warehousing and Knowledge Discovery, pp. 9–22. Springer (2014)
17. Wang, Y., Kung, L.A., Byrd, T.A.: Big data analytics: understanding its capabilities and potential benefits for healthcare organizations. *Technol. Forecast. Soc. Change* **126**, 3–13 (2018)
18. Zhang, Y., Qiu, M., Tsai, C.-W., Hassan, M.M., Alamri, A.: Health-CPS: healthcare cyber-physical system assisted by cloud and big data. *IEEE Syst. J.* **11**(1), 88–95 (2017)
19. Ziegler, P., Dittrich, K.R.: Data integration-problems, approaches, and perspectives. In: Conceptual Modelling in Information Systems Engineering, pp. 39–58. Springer (2007)



Accelerating Road Sign Ground Truth Construction with Knowledge Graph and Machine Learning

Ji Eun Kim¹(✉), Cory Henson¹, Kevin Huang¹, Tuan A. Tran²,
and Wan-Yi Lin¹

¹ Bosch Corporation Research, Pittsburgh, USA

{jieun.kim,cory.henson,kevin.huang,wan-yi.lin}@bosch.com

² Bosch Chassis Systems Control, Stuttgart, Germany

anhtuan.tran2@bosch.com

Abstract. Having a comprehensive, high-quality dataset of road sign annotation is critical to the success of AI-based Road Sign Recognition (RSR) systems. In practice, annotators often face difficulties in learning road sign systems of different countries; hence, the tasks are often time-consuming and produce poor results. We propose a novel approach using knowledge graphs and a machine learning algorithm - variational prototyping-encoder (VPE) - to assist human annotators in classifying road signs effectively. Annotators can query the Road Sign Knowledge Graph using visual attributes and receive closest matching candidates suggested by the VPE model. The VPE model uses the candidates from the knowledge graph and a real sign image patch as inputs. We show that our knowledge graph approach can reduce sign search space by 98.9%. Furthermore, with VPE, our system can propose the correct single candidate for 75% of signs in the tested datasets, eliminating the human search effort entirely in those cases.

Keywords: Knowledge graph · Meta-learning · Road sign classification · Data annotation · Crowd-sourcing · Human in the loop · Autonomous driving

1 Introduction

Recognizing and understanding road signs are important features of advanced driver-assistance systems (ADAS), which are offered in modern vehicles via technologies such as road-sign recognition (RSR) or intelligent speed adaption (ISA)¹. Recent RSR and ISA solutions heavily use Machine Learning methods and require comprehensive, high-quality datasets of road sign annotation as ground truth. To be ready for real-world usage, the ground truth must be built from test drives around the world. The number of road sign images to be

¹ These features are mandatory in all new cars sold within Europe from May 2022 [3].

annotated can be enormous, up to more than ten millions each year. Any representative sample of these images that covers enough countries and conditions will be of considerable size. It is therefore crucial to optimize the annotation task and minimize annotator’s time in each session.

There are two main challenges in performing a road sign annotation task. First, there are too many road signs to search through to find a matching one (USA alone has more than 800 federally approved road signs, and 10 states in USA have their own state conventions which are different from the federal convention [23]). This makes manual classification of each sign instance against a full palette of signs infeasible². One solution is to have a machine learning system limit the number of candidates for human annotators to search from (e.g., to 5 signs).

The second challenge lies in the fact that different countries follow different conventions on road signs. For instance, USA follows MUTCD [23], while European countries adopt the Vienna convention [9]. Some countries adopt multiple conventions, and some introduce different variants in features such as colors, fonts, size, etc. This is illustrated in Fig. 1. No annotator possesses full knowledge of all road sign systems and may choose the wrong ones, especially when the instance is not clear (e.g., gray-scale images, night images, and so forth).



Fig. 1. Examples of various road signs across countries representing different shapes, colors, icons and languages

In this work, we address the above issues with a novel solution that combines knowledge graph and machine learning to assist annotators and accelerate the ground truth annotation. The idea is that all road signs have some basic visual features, and we can navigate through the knowledge graph of these visual features (focusing on country-specific sub-graphs by using GPS data associated with the images), to locate the candidate signs, supporting the sense-making of human annotators. We show that this approach can reduce the search space for signs by 98.9%, from 845 signs to 8.92 signs. This reduction in search space translates to reduced search effort and time by human annotators for locating the correct sign. To further reduce the search space, we propose to use a Variational Prototyping-Encoder (VPE [16]) that uses one-shot learning to find matching signs, even if unseen in prior training data. To summarize our contribution, in this work we have:

² Note that we define “sign” as a prototypical sign, and differentiate it from “sign instance” which is an instance of a prototypical sign as seen in drive data. For example, if a set of images contains 4 stop signs and 4 yield signs, then there are 8 sign instances in total, and 2 signs (stop and yield).

- introduced Road Sign Ontology (RSO) to represent salient features of road signs,
- proposed crowd-sourcing techniques to construct the Road Sign Knowledge Graph at scale across countries and states
- used knowledge graph to reduce the sign search space by 98.9%
- built a VPE model that is combined with the knowledge graph to further reduce the search space. This approach can propose the correct single candidate for 75% of signs in the tested datasets even for unseen signs during the training phase, eliminating the human search effort altogether in those cases

In the following sections, we quickly summarize related work, explain the construction of the ontology and knowledge graph, and detail our knowledge graph and machine learning-assisted road sign annotation pipeline. Finally, we present our experimental results on internal and public datasets.

2 Related Work

Road Sign Detection and Classification. Due to the ubiquitous presence of road signs, detecting and classifying them are important tasks in automated driving research [25, 28]. A typical framework involves the detecting of bounding boxes containing road signs from video or image data [21], then mapping them to canonical signs, often represented by an image prototype. While much of existing work focused on the visual properties of the road signs such as color [18], shape, and templates [19, 20], the varying of such properties as regulated in different conventions is not well addressed [21]. Although there are several open datasets for road sign recognition and classification [6, 17, 25, 26], they cover only a very small portion of the signs and face several challenges when adapted to the international setting.

Using Knowledge Graph to Improve Evaluation. Our work is characterised by the usage of an ontology and knowledge graphs to assist human annotators in complex tasks. In this perspective, it has been studied in crowd-sourcing research communities, by introducing domain knowledge to support human sense-making and learning during the microtask. Early work in this direction can be found in database or text mining applications [8, 11]. For visual tasks, some open ontologies are built to help humans annotate the biomedical concepts [7]. To the best of our knowledge, the use of a large-scale knowledge graph, consisting not only of taxonomy but also of Resource Description Framework (RDF) facts curated from several sources for evaluating visual tasks in autonomous driving area is completely unexplored.

Our construction of the Road Sign Knowledge Graph is also inspired by related work in evaluating quality of large-scale linked data using crowd-sourcing [14, 24]. For images, previous work suggests that careful design of micro-task should take into account visual effects on human across countries [14].

Meta-learning for Road Signs. A recent approach for dealing with the incompleteness of road sign classes is to use meta-learning, including few-shot and one-shot learning [16, 27]. The idea is to represent each unseen road sign class with an image prototype, which is jointly encoded with existing classes, either driven by a distance metric or a meta-model, to learn to translate each real image depicting a road sign. Some work combines knowledge graph into meta-learning to improve the performance via concept propagation [15]. In autonomous driving, to the best of our knowledge, our work is the first to propose the combination of knowledge graph and one-shot learning to construct ground truth efficiently.

3 Road Sign Ontology

This section describes an ontology for representing and reasoning about road signs, namely, the Road Sign Ontology (RSO). This ontology, and its conformant knowledge graph (see Sect. 4), is used to assist in the data annotation process and the training of machine learning models for road sign classification. RSO seeks to represent the salient visual features of a road sign that are discernible through sight or imaging, and is modeled using the Web Ontology Language (OWL2) [13]. See Fig. 2 for a visualization of the primary ontology concepts.

While designing RSO, two primary requirements were considered. First, the ontology should represent the features of road signs that are beneficial to the performance of machine learning algorithms. Second, the ontology should represent concepts at an appropriate level of granularity that enables human annotators to effectively identify road signs and their visual features when looking at an image.

3.1 Road Sign Features

The primary features of a road sign to be represented include its shape, color, text, and printed icons.

Shape. RSO distinguishes between two types of shapes associated with a road sign. The most obvious is the shape of the physical plate. For example, in the United States, stop signs have an octagon shape, yield signs have a downward triangle shape, and speed limit signs have a rectangular shape. There are 11 different shapes that the physical plate of a road sign could take. The second type of shape includes geometric shapes that are printed on the physical plate. Common printed geometric shapes include arrows, circles, and diagonal lines. RSO represents nine different printed shapes.

Color. Similar to shape, RSO distinguishes between multiple different types of color associated with a road sign. Specifically, a road sign can have a foreground color, a background color, and a border color. Eleven common colors are enumerated within the ontology.

Icon. Icons are a special type of shape printed on a road sign that depict various objects. The types of objects often depicted include vehicles, people, animals, and assorted traffic infrastructure (e.g., a traffic light). Given the large number of possible distinct icons, RSO only defines a few common categories, including: animal, infrastructure, nature, person, vehicle, and other.

Text. Many road signs include printed text. Stop signs print the word *STOP*, yield signs print the word *YIELD*, and speed limit signs include both the words *Speed Limit* and a number. Rather than enumerating all possible text that may be printed on a sign, RSO allows the text of a specific sign to be annotated using an OWL Datatype Property. While RSO does not define enumerations for all possible text on a road sign, it does enable the categorization of text into various types, based on the intended meaning or use. The categories of text include: speed, height, weight, time, name, and number. As an example, the text of a speed-limit sign is identified with the *speed* category, while the text of a sign announcing entrance to a town is identified with the *name* category.

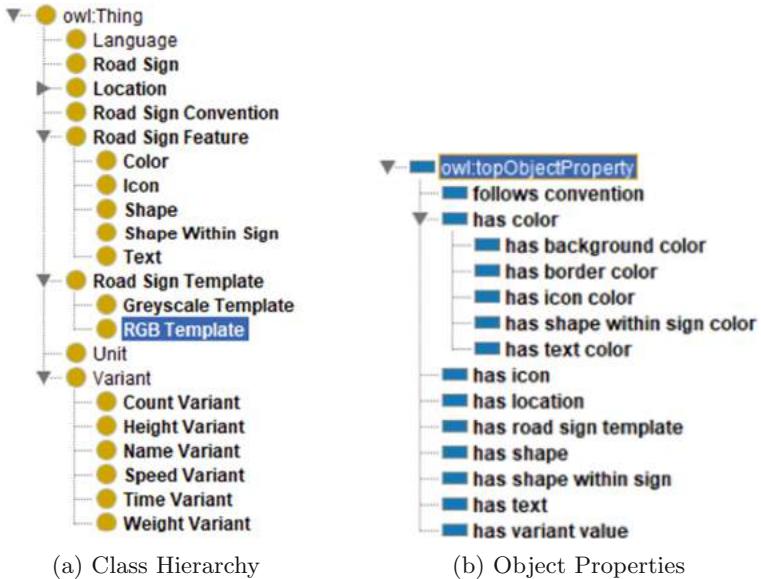


Fig. 2. Visualization of the road sign ontology with protege [22]

3.2 Road Sign Conventions and Prototypes

In practice, road signs must adhere to convention. It is this adherence that allows and empowers a person to detect and identify the meaning of a sign with only a brief glance. Such conventions define rules and constraints on how

road signs of various types should be printed and displayed. There are three primary road sign conventions in use today: the Vienna Convention [12], the MUTCD Convention [23], and the SADC Convention [4]. The Vienna Convention is used primarily in Europe and China; the MUTCD Convention (i.e., Manual on Uniform Traffic Control Devices) is used primarily in the United States; and the SADC Convention (i.e., Southern African Development Community) is used primarily in Africa. Variations of these conventions may be defined and used for more specific geo-spatial regions. For example, each state in the United States may either adhere to the federal version of MUTCD or they may define their own state-specific version. Each road sign represented by RSO may be associated with the convention to which it adheres.

Conventions also provide standard images that depict the sign. These standard images are often referred to as prototypes and provide a template for the design, construction, and illustration of signs in manuals. Prototypes often come in two versions, a full color version and a gray-scale version. RSO enables road signs to link to these prototype images on the Web.

3.3 Alignment with Domain Vocabularies

While the Road Sign Ontology defines a general vocabulary to catalog visual and conventional properties of road signs, we need to extend it to include other non-ontological structure of existing road sign datasets, and to tailor to domain-specific (ADAS) applications used in commercial systems. This is important to allow the machine learning and computer vision models, which is developed using the specific labels of the datasets or domain-specific taxonomy, to interoperate with the knowledge graphs. In our work, we integrate three open datasets: LISA [21], representing US signs, GTSRB [25], representing German signs, and TT-100K [28], representing Chinese signs. Each of these datasets have a vocabulary that is tailored to representing signs only in a specific region. In addition, we also model the taxonomy used in our different in-house annotation applications, which cover road signs of 72 countries, and are optimized for TSR function development. Below we present the high-level approach of our alignment.

To integrate these datasets, we create domain-specific vocabularies and put them under each corresponding namespace. Of course, there is a significant overlap between namespaces, but the alignment is non-trivial, for example in two following cases:

Road Sign Categories—The way that road signs are categorized is different for each domain and dataset. Each uses a distinct number of categories with divergent levels of granularity. As an example, a road sign (e.g., *truck speed limit 60*) can be mapped to a single category *truck speed limit* within one dataset (TT-100K), while mapped to both a *truck* sign category and a *speed limit* sign category in another (GTSRB).

Road Sign Features—Road sign features are also represented at different levels of abstraction within the various domains and datasets. While our special in-house annotation tools enable annotators to describe detailed features of the road signs, open datasets only use prototypical images. Furthermore, GTSRB

provides for each sign different prototypes with variants in foreground, background, and border, while TT-100k and LISA use only one prototype per each.

Figure 3 illustrates how the Road Sign Ontology is aligned with domain vocabularies. The standard `owl:equivalentClass` is used to link the sign categories, and new relations are introduced for equivalent road sign features. We also introduce the class `Variant` to embrace the different prototypes as in the case of GTSRB.

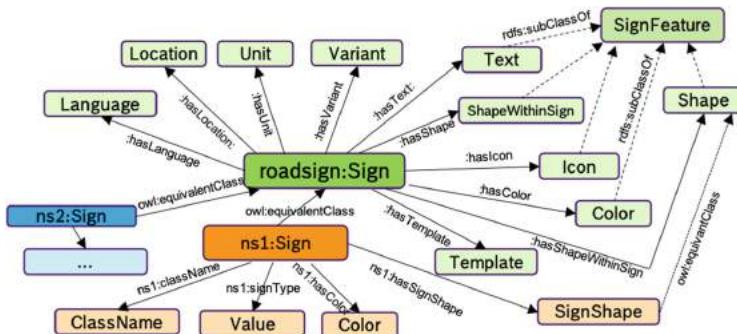


Fig. 3. Example of aligning the concept of road sign, in RSO, with concepts in other vocabularies.

4 Road Sign Knowledge Graph

This section describes our approach to build the Road Sign Knowledge Graph at scale based on RSO discussed in Sect. 3. We need to design the knowledge graph specifically to enable the recognition of road signs in different applications. However, building the knowledge graph manually is both time consuming and difficult due to lack of comprehensive domain knowledge of human annotators. In our work, we develop a two-step system: First, we rely on the crowd to construct the large-scale graphs with basic properties. Second, we align and extend the graphs to “fine-tune” to domain-specific data and vocabularies. Below we detail our approach.

4.1 Road Sign Knowledge Graph Construction with Crowdsourcing

As the number of localized road signs is considerably large and requires many people who know different languages used in different locations, we leverage crowds recruited from three online crowd-sourcing marketplaces with global registered work-forces: Amazon Mechanical Turk [1], Click Worker [2] and UpWork [5]. We design two tasks in each platform:

Identifying Road Sign Templates for Various Countries. In the first task, we ask the crowd to find a Web site that provides official road sign documents having templates of the road sign for a target country or state. We ask at least three crowd workers to get consensus on recommended sources. After identifying a source, manual template extraction is required if the identified resource does not support a separate image file format for each road sign template.

Road Sign Feature Extraction. In the second task, we create a web-based application for crowd workers to extract the sign features used for constructing the road sign knowledge graph. A snapshot of the user interface is shown in Fig. 4. In detail, the crowd worker is asked to look at a road sign template and provide the plate shape, background color, border color, additional shapes (e.g., left arrow) inside the plate, icons (e.g., vehicle), text and variants (e.g., street name) if applicable. This microtask can be done by any crowd workers and does not require road sign knowledge i.e., the meaning of a given road sign template. All answers can be selected from the provided options, except text that should be typed in the text field. Therefore, we do not specify any driving experience as qualification or require any training; we simply provide an instruction with examples and recruit workers whose approval rate is greater than 98%. However, each HIT includes one gold standard road sign, for which the system knows the ground truth in order to screen scammers who intentionally try to fool our system. Each individual sign template is presented to one worker, and one internal expert reviews the answers from the worker, followed by an additional review with another internal expert for further clarification if necessary.

We have constructed a generic road sign knowledge graph with 2,804 road signs from two different countries as of the reporting date. (The number of signs will be increased when adding signs for more countries.) In detail, 281 HITs (i.e., Human Intelligent Task) are published to Amazon Mechanical Turk when extracting sign features for U.S federal signs, three US states signs and German signs. Each HIT includes up to 10 road sign templates. The number of questions for each sign template varied based on sign features shown in each template. 7 to 11 questions are answered for an individual sign template. It took 23.7 min on average to complete one HIT. The workers provided correct answers (98%) for most questions on shapes, icons, and text. We observed that some colors such as yellow and orange are not properly annotated by a handful workers, possibly due to color representation in the screen they used. We did not reject these answers but correct the answers during the review process. Surprisingly 35% of answers are incorrect when they are asked to provide variants or units for certain signs. For instance, a no parking sign (see the last sign in Fig. 1) includes a time variant (8:30 AM to 5:30 PM), but some workers were not able to recognize this time variant. Also the ground truth for the SPEED LIMIT 30 unit is “miles per hour”, but some workers gave “miles” as the unit. 41% of workers reported that they drive everyday and 6% of workers do not have a driving license. We have not seen any differences in the road sign knowledge graph annotation quality with respect to workers’ driving experience. 1% of scammers are detected when running this batch of HITs.

Please look at the image carefully and provide an answer for each question on Right.



3. What are other geometric shapes INSIDE of the physical sign shape you answered in the 1st question? If there are many arrows, multiple choices are required.

- Arrow to indicate Forward
- Arrow to indicate Backward
- Arrow to indicate Right

3.1. What are colors of this shape?

- | | | |
|---|---------------------------------|---------------------------------|
| <input checked="" type="checkbox"/> Black | <input type="checkbox"/> Blue | <input type="checkbox"/> Brown |
| <input type="checkbox"/> Green | <input type="checkbox"/> Orange | <input type="checkbox"/> Pink |
| <input type="checkbox"/> Purple | <input type="checkbox"/> Red | <input type="checkbox"/> Yellow |
| <input type="checkbox"/> Yellow-Green | <input type="checkbox"/> White | |

- Arrow to indicate Left
- Circle or Ellipse
- Circle with diagonal line
- Circle with horizontal line
- Diamond
- Zebra Crossing
- Triangle Up
- Triangle Down
- Triangle Right
- Triangle Left
- Rectangle
- Octagon
- Pentagon

Fig. 4. Screenshot of the crowdsourcing task for road sign attribute extraction

Validated attributes for each sign template are translated into RDF facts of a corresponding entity of type `Sign` in the *generic knowledge graph*. Next, we describe how the facts are refined to produce different domain-specific graphs.

4.2 Alignment of Road Sign Knowledge Graph to Domain-Specific Knowledge Graphs

To get the knowledge graphs specific to different domains, we first create a separate graph for each domain from the relevant sub-graph of the generic knowledge graph (for instance, a sub-graph containing all facts about a target country or state). We then perform the following three alignments to extend and refine the domain-specific graphs:

- **Automated Reasoning:** As our RSO is compliant to OWL-DL, the system can perform semantic reasoning to add more facts, such as adding category to the sign via its color and shapes. The reasoning can also create facts in different granularity. For example, the property *has foreground color* can be refined to *has icon color*, *has text color*, *has shape within a sign color* using the subsumption, or mapped to a more generic property *has color*.
- **Auto-Transformation for Individual Triples:** If the content in a triple in the generic graph is transformable, we apply rules to get more facts. For instance, the text “SPEED LIMIT 30” can be transformed into two triples with “SPEED LIMIT” as a text and 30 as a numerical value in the domain-specific graph.
- **Manual Alignment:** Finally, experts of ontology alignment are also advised to add new vocabularies into the domain-specific graphs when needed. For example, the category/class names used in a domain are often acronyms, which cannot be automated without additional inputs.

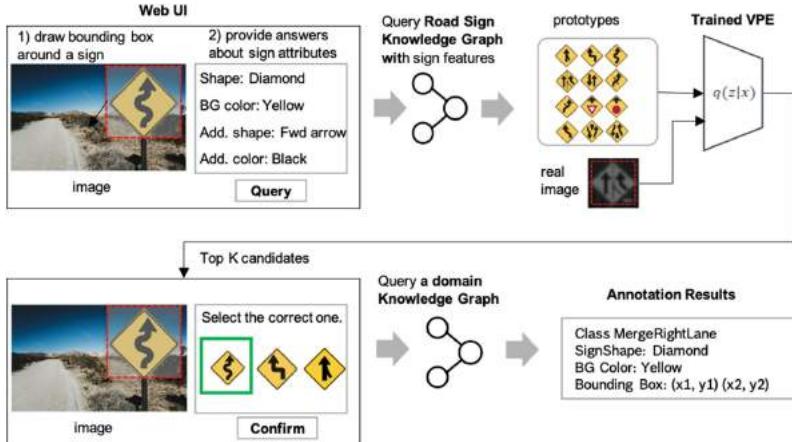


Fig. 5. Road sign annotation process

The result of this step is a generic road sign knowledge graph and multiple domain-specific graphs. The graphs with new signs are regularly reviewed by domain experts. Meanwhile, only relevant graphs are loaded into the downstream annotation task (described in Sect. 5) to mitigate implications of any remaining inconsistency.

4.3 Notes on Architecture

The knowledge graphs are stored and processed in the StarDog Enterprise Cluster³. We use a MongoDB⁴ database to store intermediate annotations and perform multiple validation before storing it in StarDog. We extended JenaBean [10] to convert the Web application data model to the triples that follow Road Sign Ontology (RSO) described in Sect. 3. We use the built-in semantic reasoning and regular expression capabilities in StarDog to perform the graph alignment.

5 Knowledge Graph and Machine Learning Assisted Road Sign Annotation Pipeline

This section describes how the proposed Road Sign Knowledge Graph and the domain Road Sign Knowledge Graph is used in our road sign annotation tool, along with a machine learning classifier, to assist a human annotator on an image.

³ <https://www.stardog.com>.

⁴ <https://www.mongodb.com>.

Tasks for the Human Annotator. As shown in Fig. 5, the task for the human annotators is to draw a bounding box around a road sign on an image and select a matching sign prototype from a small palette of signs. We do not expect the human annotators possess knowledge on traffic systems which often vary across countries. Furthermore, this simple task does not require a separate training session about road signs to complete. Instead, annotators interact with the system by providing road sign features which are visible inside of the bounding box they draw on the Web UI. The annotator provides visual attributes such as plate sign shape and background color as common attributes, and icons, text and additional shapes as optional attributes. Then, they are asked to select a sign template from the ordered candidates. The task execution time and quality depends on the search space, i.e., the number of candidates they have to visually compare, and the quality of the image. If a road sign is not exactly matching with any candidates, then a worker selects a most matching sign or a default sign template that represents common attributes such as plate shape and background color with an indication of missing a sign template. This indication is further used by the system to add missing signs to the knowledge graph. Note that road signs in the wild are sometimes customized and knowledge graph constructed based on the official documents cannot cover all possible variation at the beginning of the system deployment. Rather we support to scale up while running the system with workers' feedback.

Use of Road Sign Knowledge Graph and Machine Learning Model. Our tool supports the human annotator by providing a handful of road sign candidates that match the attributes given through a knowledge graph search. If the number of sign candidates is greater than a threshold K , then the machine learning model is applied to further reduce the number of candidates.

We integrate the Variational Prototypical Encode (VPE) model [16] to predict $top-K$ road sign candidates. The inputs for the VPE model are: 1) a cropped image patch around the bounding box that the annotator draws on the real road image and 2) sign templates filtered by the Road Sign Knowledge Graph. These two inputs are encoded into the latent space, and nearest neighbor classification is used to rank road sign templates. The system returns top K candidates back to the human annotator.

Prediction of unseen classes is crucial in the road sign annotation due to rare road signs in the long tail. Model prediction of conventional classification with known classes cannot assist human annotators effectively unless that model is trained with large datasets that include rare classes. The encoder in the training phase of VPE encodes real images to latent distribution and the decoder reconstructs the encoded distribution back to a prototype that corresponds to the input image. By doing so, the trained encoder is used as a feature extractor and VPE learns image similarity and prototypical concepts instead of learning classification itself. Therefore, the pre-trained model can predict novel classes that are unknown during the training time.

Querying the Domain Knowledge Graph and Human Validation. Upon the final selection of a matching road sign prototype, the system queries our domain specific knowledge graph to get relevant attributes needed for annotation, such as a corresponding class name, class description, colors, sign shape and text along with a coordinate of the bounding box.

6 Evaluation

This section provides experimental results to validate our assumption that the knowledge graph and VPE model can reduce the number of sign templates which the annotator has to search through to create ground truth.

6.1 Search Space Reduction with Road Sign Knowledge Graph

First we discuss the search space reduction from the knowledge graph alone.

Dataset Used in the Evaluation. Our first evaluation uses 51,000 frames of road scenes recorded from one of front cameras mounted in our testing vehicles. The recorded frames cover various cities in US. A total of 6,253 road signs instances are observed. 50 road sign classes are annotated in this dataset. Note that a road sign class in this data model can have multiple sign templates if the meaning of signs are the same in the application. For example, SPEED LIMIT 30 and SPEED LIMIT 50 have the same road sign class with a different value (30 and 50 respectively) for each sign. Some road signs that are not relevant to the application are modeled as other classes.

Knowledge Graph Used in the Evaluation. The Road Sign Knowledge Graph constructed with 845 US federal MUTCD sign templates are used for this experiment. We select only one sign template for each class and run SPARQL queries. Common query parameters for all signs are plate shape and background color. Shape within a sign, icon and text are used optionally depending on the sign template.

Evaluation Results. To understand the distribution of signs on the graph, we run queries with common parameters (shape and background color) on the graph database. The query results show that 42% of signs have rectangle shape and white background color, and 14% of signs have diamond shape and yellow background color.

We run 50 queries with common and optional query parameters to evaluate the search space reduction for observed sign classes in our dataset. Table 1 shows the distribution of the search space size range used from the evaluation. The results show that the size of the search space is reduced by 98.9% (from 845 to a mean of 8.92 signs) on average when querying road signs with minimum 3 and maximum 5 attributes. The reduced number of road sign candidates should decrease the efforts for the human annotator because the number of comparison against a real sign is reduced proportionally.

Table 1. Search space reduction distribution

Search space size	Percentage (%)	Mean	Stdev
1 to 5	38%	8.92	7.36
6 to 10	16%		
11 to 15	20%		
16 to 20	12%		
21 to 25	14%		

Training Time and Learning Curve. In addition, human annotators spend weeks to get trained on road sign annotation for various countries to be able to make intelligent decisions when they annotate signs that are different from provided road sign templates, according to our interviews with annotation experts. Uncertainty in classifying variants of similar road signs creates either failure in the ground truth annotation or requires additional rigorous reviews relying on the trained domain expert. Training time and learning curve can be improved by increasing the number of road sign prototypes via crowd-sourced knowledge graph construction.

6.2 Variational Prototyping-Encoder Model Accuracy with Respect to Search Space Size

The VPE re-ranks the sign candidates filtered by the knowledge graph. Our second hypothesis is that the accuracy of VPE can be improved with the reduced number of candidate classes to the encoder during inference. This improvement can further increase the efficiency of the human annotators by further reducing the search space.

Datasets used in the Evaluation. We validate this hypothesis on two benchmark road sign datasets to evaluate how the reduction of the search space from knowledge graph improves the classification accuracy: 1) German Traffic Sign Recognition Benchmark (GTSRB) [25], containing 43 classes of road signs used in Germany where they adopt the Vienna convention. This dataset provides approximately 50,000 instances of image patches with sizes varying from 15×15 pixels to 250×250 pixels. The dataset also provides road sign template images. 2) LISA [21] dataset, containing 47 US sign classes of 7,855 annotation from the Manual on Uniform Traffic Control Devices (MUTCD) convention. The LISA dataset does not contain the road sign template images; we used the road sign templates from our Road Sign Knowledge Graph.

Training and Testing Process. We selected these datasets because significant differences in signs between these two countries allow us to evaluate the effectiveness of our knowledge graph in the VPE model when testing the model performance with road sign classes not used during the training phase. We first

trained VPE with the GTSRB dataset. The input of the VPE model is cropped image patches from the real road scenes and the output of the VPE model is the road sign template for the correct class. The VPE learned image similarity represented in embedding (vector size of 300) as well as prototypical concepts of the road sign during the training phase.

10 signs classes having the diamond shape and the yellow background color in the LISA dataset are used to test the performance by changing the number of road sign templates to the pre-trained encoder. No signs in GTSRB have the diamond shape and yellow background color. The performance covers unseen sign classes with different features during the training time. The output of the pre-trained encoder is the ranked list of road sign templates. We calculated accuracy of top 1, top 3, and top 5 ranks (the ground truth template is listed in top 1, top 3, and top 5 ranks).

Evaluation Results. The evaluation results shows that the accuracy of the model is increased by 10–13% when the search space size is reduced from 30 to 10 (see Table 2). The results also demonstrate that the model predicts the ground truth template for 75% of the dataset when the search space size is 10. In these cases, the human annotator would not need to spend any time searching for a matching sign, thus further reducing annotation effort.

Table 2. VPE Performance evaluation with different search spaces tested with signs having diamond shape and yellow background color

Search space sizes	Accuracy (top-1)	Accuracy (top-3)	Accuracy (top-5)
10	0.73	0.85	0.90
20	0.69	0.80	0.85
30	0.60	0.73	0.80

7 Conclusion

We proposed to use Road Sign Knowledge Graph for the data annotation tool for road sign annotation along with a meta-learning model. The ontology model for the road signs is intended to map to various domains and applications. We demonstrate that the construction of the Road Sign Knowledge Graph can be made scalable for signs across states and countries by using a crowd-sourcing approach. The evaluation results show that the search space for human annotators can be reduced by 98.9%, from a palette of 845 signs to only 8.92 signs, reducing human effort and easing the learning curve for road sign classification. Furthermore, the results show that integrating machine learning with the knowledge graph can produce an exact match for 75% of signs, eliminating human classification effort entirely in those cases.

References

1. Amazon Mechanical Turk. <https://mturk.com>. Accessed May 2020
2. ClickWorker. <https://www.clickworker.com>. Accessed May 2020
3. Regulation (EU) 2019/2144 of the European Parliament and of the Council. <https://eur-lex.europa.eu/eli/reg/2019/2144/oj>. Accessed May 2020
4. South African Road Traffic Signs Manual. <https://www.transport.gov.za/>. Accessed June 2020
5. UpWork. <https://www.upwork.com/>. Accessed May 2020
6. Belaroussi, R., Foucher, P., Tarel, J.P., Soheilian, B., Charbonnier, P., Paparoditis, N.: Road sign detection in images: a case study. In: 20th International Conference on Pattern Recognition, pp. 484–488. IEEE (2010)
7. Bukhari, A.C., Nagy, M.L., Krauthammer, M., Ciccarese, P., Baker, C.J.: BIM: an open ontology for the annotation of biomedical images. In: ICBO (2015)
8. Chu, X., et al.: A data cleaning system powered by knowledge bases and crowd-sourcing. In: Proceedings of the ACM SIGMOD International Conference on Management of Data (2015)
9. Inland Transport Committee: Convention on Road Traffic. Economic Commission for Europe, Vienna, Austria (1968)
10. Cowan, T.G.: JenaBean: easily bind JavaBeans to RDF. IBM DeveloperWorks (2008)
11. Dojchinovski, M., Reddy, D., Kliegr, T., Vitvar, T., Sack, H.: Crowdsourced corpus with entity salience annotations. In: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016) (2016)
12. United Nations. Economic Commission for Europe. Transport Division. Convention on Road Traffic of 1968 and European Agreement Supplementing the Convention (2006 Consolidated Versions) (2007)
13. Hitzler, P., Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S., et al.: OWL 2 web ontology language primer. W3C Recomm. **27**(1), 123 (2009)
14. Jou, B., Chen, T., Pappas, N., Redi, M., Topkara, M., Chang, S.F.: Visual affect around the world: a large-scale multilingual visual sentiment ontology. In: Proceedings of the 23rd ACM International Conference on Multimedia, pp. 59–168 (2015)
15. Kampffmeyer, M., Chen, Y., Liang, X., Wang, H., Zhang, Y., Xing, E.P.: Rethinking knowledge graph propagation for zero-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
16. Kim, J., Oh, T.H., Lee, S., Pan, F., Kweon, I.S.: Variational prototyping-encoder: one-shot learning with prototypical images. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
17. Larsson F., Felsberg M.: Using Fourier descriptors and spatial models for traffic sign recognition. In: Scandinavian Conference on Image Analysis, pp. 238–249. Springer (2011)
18. Lopez, L.D., Fuentes, O.: Color-based road sign detection and tracking. In: Image Analysis and Recognition, pp. 1138–1147. Springer (2007)
19. Loy, G., Barnes, N.: Fast shape-based road sign detection for a driver assistance system. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 70–75. IEEE (2004)
20. Malik, R., Khurshid, J., Ahmad, S.N.: Road sign detection and recognition using colour segmentation, shape analysis and template matching. In: 2007 International Conference on Machine Learning and Cybernetics, vol. 6, pp. 3556–3560. IEEE (2007)

21. Mogelmose, A., Trivedi, M.M., Moeslund, T.B.: Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. *IEEE Trans. Intell. Transp. Syst.* **13**(4), 1484–1497 (2012)
22. Musen, M.A.: The protégé project: a look back and a look forward. *AI Matters* **1**(4), 4–12 (2015)
23. Texas MUTCD. Manual on uniform traffic control devices. Texas Department of Transportation, Austin (2006)
24. Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., Lehmann, J.: Crowd-sourcing linked data quality assessment. In: International Semantic Web Conference (2013)
25. Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The German traffic sign recognition benchmark: a multi-class classification competition. In: International Joint Conference on Neural Networks (2011)
26. Timofte, R., Zimmermann, K., Van Gool, L.: Multi-view traffic sign detection, recognition, and 3D localisation. *Mach. Vis. Appl.* **25**(3), 633–647 (2011). <https://doi.org/10.1007/s00138-011-0391-3>
27. Zhou, S., Deng, C., Piao, Z., Zhao, B.: Few-shot traffic sign recognition with clustering inductive bias and random neural network. *Pattern Recognit.* **100**, 107160 (2019)
28. Zhu, Z., Liang, D., Zhang, S., Huang, X., Li, B., Hu, S.: Traffic-sign detection and classification in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)



Adjusted Bare Bones Fireworks Algorithm to Guard Orthogonal Polygons

Adis Alihodzic^{1(✉)}, Damir Hasanspahic¹, Fikret Cunjalo¹,
and Haris Smajlovic²

¹ University of Sarajevo, Sarajevo, Bosnia and Herzegovina
`{adis.alihodzic,fcunjalo}@pmf.unsa.ba, damir.hasanspahic@gf.unsa.ba`

² University of Victoria, Victoria, Canada
`hsmajlovic@uvic.ca`

Abstract. With the growing demand for public security and intelligent life as well as the expansion of the Internet of Things (IoT), it is indispensable to make a plan how to place the minimum number of cameras or guards to achieve secure surveillance. The optimal cameras placement is a hard combinatorial problem, and it can be formulated as seeking the smallest number of guards to cover every point in a complex setting. In this article, we propose an adjusted version of the bare bone fireworks algorithm and one deterministic technique for tackling cameras placement problem. Both versions of novel algorithms have been implemented and tested over two hundreds of randomly generated orthogonal polygons. According to the outcomes presented in the experimental analysis, it can be noticed that the first approach based on metaheuristics beats the deterministic method for practically all instances.

Keywords: Computational geometry · Visibility · Camera placement · Internet of Things · Swarm optimisation · Bare Bones fireworks algorithm

1 Introduction

Internet of Things (IoT) in the modern world presents the growing and enlargement of Internet technologies. The IoT links objects and actualises information distribution by employing routines such as recognition, perception, and communication. The improvement and employment of IoT have brought great benefit to people's daily life. Many technologies in the IoT rely on sensors and cameras networks since they are the hardware foundation for acquiring processing and transmitting the information. Sensor networks have employed in different branches such as the structural health monitoring [4], multimedia big data communications [16], target localisation, identification, tracking [34], robot motion planning [10], and so on. On the other side, the camera placement problem has a very significant role in IoT and computer vision tasks, and a suitable cameras network organisation can better maintenance assignments such as recognition and

tracking [18]. Also, it presents a topic in robotics where cameras are adopted as a means of acquiring information for robot navigation and direction. One exciting application related to the optimal camera placement is how to deploy the camera network on a real-world campus which is presented in the form of an orthogonal polygon. Also, traffic jam monitoring can be modelled as a problem of the optimal camera deployment to surveillance a specific orthogonal area in order to monitor numerous events on the road.

The optimal camera placement problems have been analysed for decades. The earliest paper can be traced back to the Art Gallery Problem (AGP) in the field of computational geometry [8]. The art gallery problem (AGP) dates back to the 1970s, and it was one of the earliest and most widespread problems in computational geometry. In computational geometry, it presents a visibility problem of placing at least one security guard to cover every area of a museum or gallery [9]. The AGP is focused on how to being covered the interior of a polygonal environment with the smallest number of agents such as guards, cameras, robots, sensors, where agents are usually represented in the form of vectors. In the original form, the AGP is an intractable NP-hard problem which was based on determining smallest number of security guards sufficient to see every point in an n -sided two-dimensional polygon P with or without holes. The scientists such as O'Rourke and Supowit, Lee and Lin, Katz and Rpoisman, Schuchardt and Hecker have shown that the process of looking for the smallest number of guards who can surveillance any polygons (ordinary or orthogonal) still presents an intractable NP-hard problem [19, 20, 28, 29]. In 1975, Chvátal proved that only $\lfloor \frac{n}{3} \rfloor$ cameras are sometimes necessary and always sufficient to being covered the simple polygons composed of n vertices [8]. For n -sided polygon P with h holes, O'Rourke showed that it is necessary at most $\lfloor \frac{n+2h}{3} \rfloor$ vertex guards. On the other hand, Bjorling-Sachs, Souvaine, Hoffmann and others have shown that it is quite enough $\lfloor \frac{n+h}{3} \rfloor$ guards to being covered polygons with n vertices on the outer boundary which contain h holes inside them [5, 15]. Recently, more attention was transmitted to an essential variation of the traditional art gallery problem, which is called Orthogonal Art Gallery Problem (OAGP). This problem deals with the orthogonal polygons with and without holes. Generally, the orthogonal polygons can be efficiently used to model real objects (houses, buildings, etc.) since they mostly have an orthogonal structure. An orthogonal polygon without holes is a simple polygon whose edges are either horizontal or vertical. Due to its simplicity, the modelling of an art gallery with a simple orthogonal polygon allows us to produce more capable algorithms and beautiful results. At the OAGP, the guards are usually being placed on vertices of a polygon boundary. The author's Khan, Klawe, and Kleitman showed that $\lfloor \frac{n}{4} \rfloor$ guards are sufficient and sometimes necessary to cover any simple orthogonal polygon with n vertices [17]. Three years later, Lee and Lin proved that $\lfloor \frac{n}{4} \rfloor$ guards are the exact bound for monitoring the interior of a simple orthogonal polygon composed of n vertices [20]. By applying the convex quadrilaterlization algorithm to an orthogonal polygon with holes, O'Rourke et al. had shown that $\lfloor \frac{n+2h}{4} \rfloor$ vertex guards are sufficient to guard any orthogonal polygon with h holes and n vertices [26]. The researchers Györi, Hoffmann, Kriegel and Shermer

have been shown that for the orthogonal polygons with h holes always is sufficient $\lfloor \frac{3n+4h+4}{16} \rfloor$ guards to being covered [14]. By using the colouring technique which has been used by Fisk [12] to prove the Chvátal result, the authors Avis and Toussaint have developed $O(n \log n)$ time complexity algorithm for camera placement in a simple polygon. However, the number of cameras is not minimal. Also, BJORLING-SACHS and SOUVAINÉ [5] proposed an $O(n^2)$ time algorithm for non-optimal guards positioning in a polygon P with h holes.

According to the extensive and diverse deployment of camera networks, the focus of optimal camera placement problem has gradually moved from general theoretical analysis to practical utilisation. Also, most camera placement researches were focused on the formulation of this problem, as we can see in the papers [2, 3, 6, 13, 37]. However, the fashionable investigation in automated camera placement has directed on two main objectives: taking into consideration particular user specifications or producing an efficient optimisation approach. In this paper, we have concentrated on the design of efficient optimisation algorithms for tackling interior coverage (IC) problem of the orthogonal galleries. In the literature, the linear programming methods, such as the Binary Integer Programming (BIP) are most often used to address camera placement problem, when it comes to polygons with a smaller number of vertices. Although these techniques provide exact positions for optimal camera placement problem, they are practically unusable for large-scale problems because they are unable to generate a solution in a real-time. Namely, branch and bound algorithm, which is utilised in almost all camera placement strategies to solve BIP formulations directly, has exponential complexity in the worst-case [25]. Therefore, in order to adequately addressed the large search space of a reasonably sized camera placement problem, swarm intelligence techniques such as Genetic Algorithm (GA) [11], Simulated Annealing (SA) [24], Particle Swarm Optimization (PSO) [35], Artificial Bee Colony (ABC) [7] were employed to deliver suboptimal camera positions in a reasonable amount of time.

In this paper, we suggest two classes of algorithms such as a suboptimal deterministic visibility algorithm as well as a discrete version of the bare bones fireworks algorithm (BBFWA) adjusted for tackling the camera placement problem (ABBFWA). The bare bones fireworks algorithm (BBFWA) is a minimalist global optimiser designed in 2018 [23]. Since optimal camera placement is NP-hard problem, and the BBFWA is a global minimiser, it can be expected that it will intensify exploring for sub-optimal solutions as well as accomplish fast convergence and decrease the CPU processing time. Before we apply the mentioned algorithms to tackle interior coverage of an orthogonal polygon, we should exploit preprocessing techniques for the polygon decomposition into convex components, which will be covered by a minimal number of cameras. In order to show the power of proposed techniques, both algorithms have been tested on 215 various randomly generated simple orthogonal polygons. The results produced in the experimental analysis were compared with the one reached by our sub-optimal deterministic algorithm, which we also have been implemented for comparison purposes. From the experimental results, it can be seen that our first stochastic ABBFWA approach is a better technique and yields fittest solutions in a reasonable amount of time.

The structure of the article is as follows. Problem formulation for IC of a simple orthogonal polygon is described in Sect. 2. In Sect. 3, we propose a suboptimal deterministic visibility algorithm. The details of our adjusted bare bones fireworks algorithm (ABBFWA) are presented in Sect. 4. Experimental and comparative results of applying our algorithms are presented in Sect. 5. Finally, conclusions and suggestion for future work are discussed in the last section of the paper, Sect. 6.

2 Mathematical Problem Definition

In this section, we tackle the OGP in which for a given simple orthogonal polygon P which bounds an art gallery or any other object (e.g. an image of campus, an airport environment, an environment for monitoring of traffic jam, etc.), the main aim of a visibility algorithm is to ascertain a sub-optimal number of guards or cameras required to make its interior covering. Foremost, we will briefly introduce some additional notation to facilitate exposure. For any two distinct points v_1 and v_2 in the plane, we denote by $\overline{v_1v_2}$ the segment whose two endpoints are v_1 and v_2 . A planar polygon P presents a closed plane figure whose boundary is composed of segments $\overline{v_iv_{i+1}}$ ($i = 0, 1, \dots, n-1$), where $v_n = v_0$. Also, a polygon P is simple if it is not self-crossing and has no holes. A vertex v of a polygon P is reflex if the internal angle at v is greater than 180° . A planar polygon P is a simple orthogonal polygon (without holes) if its edges are parallel to the x or y axis. A planar polygon P is concave (non-convex) if there are two points u and w inside of P such that the segment \overline{uw} is not entirely contained in the P . A planar polygon P is convex if it is not concave. Also, a concave polygon must have at least four sides, and it always has at least one reflex interior angle, that is, an angle with a measure that is between 180° and 360° exclusive. Any point u in P is said to be visible from any other point w in P if and only if the segment \overline{uw} does not intersect the exterior of P as well it is entirely contained in P . For any point $u \in P$, the set of all points in P which are visible from a vertex u is called the visibility region of u , and we denote that set by $F(P, u)$. If the point u is a vertex of the polygon P , i.e. exists some index $k \in \{1, 2, \dots, n\}$ such that $u = v_k$, then we call the subset $F(P, u)$ of P fan F_k , where the vertex v_k denotes the fan vertex of the set F_k . On the other hand, let u is not a vertex of the polygon P . Then, the set $F(P, u)$ is called a region under surveillance from the point u . In order to perform the coverage of the interior by using the optimal number of cameras, we should make polygon decomposition of an orthogonal polygon P into a set of nonoverlapping convex parts C_j such that their union is the entire region of P . There are several ways how to accomplish dividing a simple orthogonal polygon into non-overlapping convex sub-polygons or components [27]. In this paper, the partitioning of a polygon into convex parts has obtained by exploiting triangulation. To efficiently perform triangulation, we have implemented a very efficient algorithm whose time complexity is proportional to the $O(n \log n)$ [9]. This algorithm consists of two steps. In the first step, we made a partition of an orthogonal polygon with n vertices into monotone pieces in $O(n \log n)$

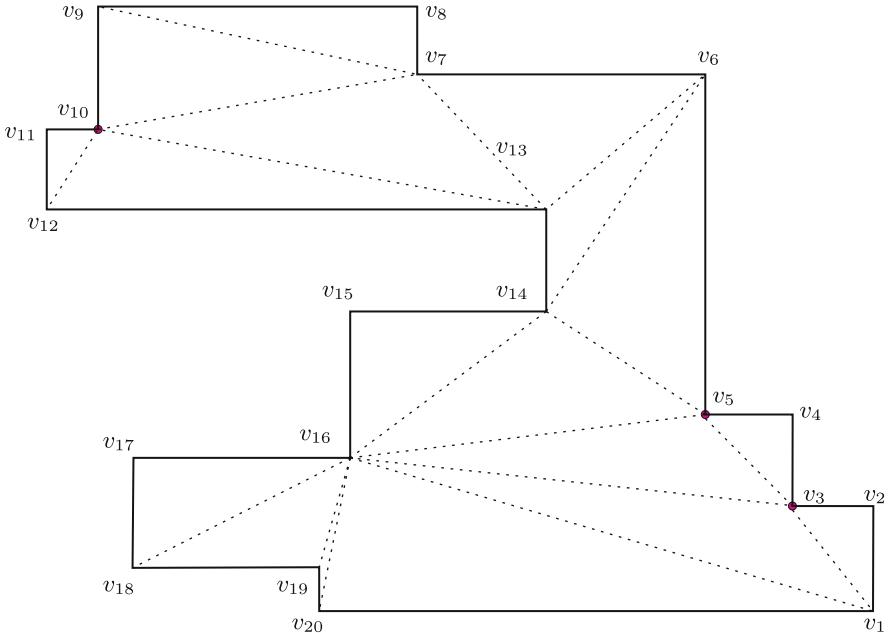


Fig. 1. The interior of the orthogonal polygon has covered by vertices v_3 , v_5 , and v_{10}

time, while in the second step, we triangulated monotone pieces (polygons) in linear time $O(n)$. The above steps together imply that any orthogonal polygon P can be triangulated in $O(n \log n)$ time. Since a triangulation of a polygon P composed of n vertices produces $n - 2$ triangles, we conclude that the number of components is equal $n - 2$. By introducing these components, optimal camera placement to perform IC of an orthogonal polygon P has reduced on seeking the smallest number of cameras which can see all components. In order to determine those cameras, we will first create $n - 2$ components C_1, C_2, \dots, C_{n-2} , so that arbitrary fan F_j ($j \in \{1, 2, \dots, n\}$) contains the indexes of components which are visible from the vertex v_j . In the following, we consider the creating of fans for the orthogonal polygon P composed of 20 vertices as it depicted in Fig. 1. It's not hard to notice that for the vertex v_1 , a fan F_1 has the indexes 1, 2, 3, 4, 5, 6, 7, because from the vertex v_1 , it can be seen the components (triangles) such as C_1 (the triangle $\triangle v_1 v_{16} v_{20}$), C_2 (the triangle $\triangle v_1 v_3 v_{16}$), C_3 (the triangle $\triangle v_1 v_2 v_3$), C_4 (the triangle $\triangle v_{16} v_{19} v_{20}$), C_5 (the triangle $\triangle v_3 v_5 v_{16}$), C_6 (the triangle $\triangle v_5 v_{14} v_{16}$) and C_7 (the triangle $\triangle v_{14} v_{15} v_{16}$). The contents of the other fans can be determined quite analogously. As we can witness from Fig. 1, exactly three cameras (for example, some synthetic omnidirectional cameras whose depth of field (DoF) or maximum viewing distance is enough large) v_3 , v_5 , and v_{10} were enough to being achieved optimal camera placement, because all triangles will be covered if we put those cameras in the mentioned vertices.

3 Visibility Algorithm for IC of the OGP

In this part, we will explain in detail the deterministic visibility algorithm for interior covering of an orthogonal polygon P . The deterministic version of our visibility algorithm will determine a suboptimal number of cameras needed to being visited all components of the polygon P . In this paper, the positions of the cameras determined by the vertices of the polygon. By taking into account the above-introduced definitions, the main idea of our visibility method has summarized by Algorithm 1. At the beginning of the algorithm, we determine such fan F_{i_1} , which contains the largest number of component indexes covered by the vertex v_{i_1} . In that case, we save the number i_1 as the index of the first camera which covers a specific interior of a polygon. After that, we update the remaining sets F_j by removing from them all the elements which appear in the set F_{i_1} , i.e. we make difference $F_j \leftarrow F_j \setminus F_{i_1}$ for all fans. It implies that the set F_{i_1} becomes empty, so it is no longer considered. For non-empty updated fans F_j , we repeat the same procedure as at the beginning of the algorithm, i.e. we select the fan F_{i_2} which has the most elements, and then take that index i_2 be the index of the second camera. It is clear now that the camera with index i_1 covers more components than the camera with the index i_2 . By repeating the mentioned procedure, we can note that after a certain number of iterations, all fans F_i will be empty, which is an indicator for the end of the algorithm. Now, the generated numbers i_1, i_2, \dots, i_k present the indexes of vertices from which the entire interior of the polygon P can be seen, where k denotes the number of cameras required to being covered the interior of the orthogonal polygon P . From the pseudo-code presented in Algorithm 1, we can see that the method stops when all fans become empty. In other words, since the union of fans F_i ($i = 0, 1, \dots, n - 2$) denotes the indexes of components from which the original polygon P is composed, it is easy to conclude that the algorithm ends as soon as all components are covered.

4 Bare Bones Fireworks Algorithm

Fireworks algorithm (FWA) is a prominent swarm intelligence algorithm proposed by authors Tan and Zhu in 2010 [30]. The FWA sparked by the process named fireworks explosion. Since the original version of the FWA had certain shortcomings, the authors put an effort to follow the growth on the algorithm provided by the other researchers, and as a result, they locate the drawbacks and eliminate them. The outcome of their work is several variants of the algorithm that were utilised to various problems. The first improved version is the enhanced fireworks algorithm (EFWA) proposed in 2013 as a solution to the problems seen in the first version of the FWA [30]. EFWA algorithm launched five corrections that encouraged better exploration and exploitation. The EFWA was accompanied by a dynamic and adaptive search proposed in 2014 [21, 38]. Further enhancements of searchability of the FWA was gathered in the fireworks with covariance mutation [36]. Finally, a cooperative framework for FWA has given in 2015 [39]. Two of the latest versions are guided fireworks algorithm

Algorithm 1. Visibility Deterministic Algorithm for Interior Coverage of OGP

-
- 1: Set $n_c \in 0$, $L_C \in \emptyset$, where n_c is a number of cameras and L_C is their list.
 - 2: Determine a triangulation of n sided orthogonal polygon P . Let us denote the obtained triangles as components C_1, C_2, \dots, C_{n-2} .
 - 3: For each vertex v_i ($i = 0, 1, \dots, n - 1$) determine the fan F_i by adding indexes of components C_j into F_i which are completely visible from the vertex v_i .
 - 4: Initialize the list of all fan's indexes with $F \in \{F_0, F_1, \dots, F_{n-1}\}$.
 - 5: **while** $n_c \neq n$ **do**
 - 6: From the list F , find the fan that has most elements and denotes its index by i .
 - 7: Put to the list L_C the camera $v_i \in P$ which was referred to the biggest founded fan F_i from the previous step.
 - 8: From all fans F_j remove the elements which were appeared to the set F_i , i.e. $F_j \in F_j \setminus F_i$.
 - 9: Set $n_c \in n_c + |F_i|$ and remove the fan F_i from the list F .
 - 10: **end while**
-

proposed in 2017 [22] and bare bone fireworks algorithm (BBFWA) from 2018 [23]. Even though the versions were updated in the short term, each of versions was used for different purposes in numerous fields such as image processing [33], machine learning [32], etc.

Algorithm 2. Pseudo-Code of Bare Bones Fireworks Algorithm (BBFWA)

-
- 1: Generate randomly the initial solution \mathbf{x} from the interval (Lb, Ub) , where Lb and Ub are lower and upper boundaries for the solutions, respectively.
 - 2: Apply the objective function f to evaluate the solution \mathbf{x} , and for whole population P composed of n agents (sparks), set the initial size of hypercube as $A = Ub - Lb$.
 - 3: **while** termination criteria is not met **do**
 - 4: For each agent $\mathbf{s}_i \in P$ ($i=1,2, \dots, n$), randomly draw its components s_{ij} from the interval $(x - A, x + A)$ by using a uniform distribution. Also, apply the mapping operator to each generated solution \mathbf{s}_i as well as evaluate its fitness function $f(\mathbf{s}_i)$.
 - 5: Among all generated solutions \mathbf{s}_i ($i=1,2, \dots, n$), find the smallest fitness value as f_j . If $f_j < f(\mathbf{x})$ then update the current best solution \mathbf{x} with $\mathbf{x} \in \mathbf{s}_j$ and set $A \in C_a \cdot A$. Otherwise, keep the old best solution \mathbf{x} as the best one and set $A \in C_r \cdot A$.
 - 6: **end while**
 - 7: Return the best found solution \mathbf{x} .
-

The bare bones fireworks algorithm (BBFWA) is a very simplified version of fireworks algorithm whose pseudo-code is given in Algorithm 2. This is absolutely a significant simplification compared to the previous versions of the FWA. Although the BBFWA is the simplest version of FWA, it is also a state-of-the-art version considering the outcomes shown in paper [23]. In the BBFWA, the number of fireworks was fixed to one, which implies that strictly one agent \mathbf{x} , i.e. the fittest one, is being transferred through iterations. The constant number

of sparks or solutions \mathbf{s}_i ($i=1, 2, \dots, n$) is produced around the firework \mathbf{x} , and they are utilised for exploring the solution space. As we can see in Algorithm 2, all solutions \mathbf{s}_i ($i = 1, 2, \dots, n$) are generated inside the hyper-rectangle in d -dimensional space of the side A , where d is dimension of the considered problem, and A is the algorithm's parameter. By using parameter A , exploration and exploitation are being realised by adjusting the dimension d of the hypercube around the fittest solution \mathbf{x} . Namely, the small values of the parameter A encourage space exploitation, which is desirable in cases when the satisfactory solution is found, and it is expected that the encouraging area is found if the fittest solution \mathbf{x} remains unchanged for two generations. While the best solution \mathbf{x} continues to be the same, parameter A is decreased by the first algorithm's parameter $C_r < 0$. Otherwise, if the best solution \mathbf{x} is being updated between two iterations, then it is thought that the solution space is not explored enough, so the size of the hypercube was grown by the second algorithm's parameter $C_a > 1$.

5 Optimal Camera Placement by ABBFWA

In this subsection, we provide in detail the adjusted version of the discrete bare bones fireworks algorithm (ABBFWA) used for the optimal camera placement problem to tackle IC of the orthogonal polygons. Before customizing the discrete version of the BBFWA to solve IC of the polygons, we will first conduct fan preprocessing based on their elimination to increase in the performance of the algorithm, additionally. Namely, if an arbitrary vertex i can see all the components C_j contained in the fan F_j which can visit the vertex j , then it is said that the fan F_i covers the fan F_j , so fan F_j can be removed. It directly impacts on the reducing size of fans. The main problem for all swarm intelligence techniques is how to design an agent for a particular type of problem. Specifically in this article, an agent is defined as a vector $\mathbf{x} = (x_{i_1}, x_{i_2}, \dots, x_{i_d})$ of dimension d , where the binary coordinates x_{i_k} are related to indices of the reduced fans, while d is a number of fans after their elimination. For example, for the orthogonal polygon P in Fig. 1, a decision vector \mathbf{x} has four elements after elimination $\mathbf{x} = (x_3, x_5, x_{10}, x_{13})$, and the ABBFWA should generate for such polygon P the following optimal solution: $\mathbf{x} = (1, 1, 1, 0)$. This further means that fans F_3 , F_5 and F_{10} cover the interior of polygon P by using the components (triangles) contained in them. Another problem that arises here is how to treat the agents that are unable to cover the interior of a polygon. For example, for the orthogonal polygon in Fig. 1, if the algorithm generates the solution $\mathbf{x} = (1, 0, 0, 1)$, then the components F_3 , and F_{13} do not cover interior of the polygon. We will call this solution an infeasible solution. To enable the algorithm to operate simultaneously both with feasible and infeasible solutions, we apply Deb's rules in this paper, analogously as we applied them in the paper [1]. Thus, at the beginning of the algorithm, none type of solution is preferred, while at the end of the algorithm, only feasible solutions with a minimum number of non-zero elements (ones) are taken into consideration.

Algorithm 3. Adjusted Bare Bones Fireworks Algorithm (ABBFWA) for OGP

-
- 1: Determine a triangulation of of n sided orthogonal polygon P . Let us denote the obtained triangles as components C_1, C_2, \dots, C_{n-2} .
- 2: For each vertex v_i ($i = 0, 1, \dots, n - 1$), determine the fan F_i by adding indexes of components C_j into F_i which are completely visible from the vertex v_i .
- 3: Create zero-one matrix $A = (a_{ij})$ of $n \times n - 2$ such that $a_{ij} = 1$ if the fan F_i contains the index j , i.e. if vertex i covers the component (triangle) C_j . After that, make preprocessing and reduce the number of fans or rows of the matrix A .
- 4: Set dimension d to be a number of the fans after elimination. Also, set $Lb = 0$, $Ub = 1$, $A = Ub - Lb$, and initialize the vector \mathbf{x} of dimension d to ones, i.e. to cover all fans.
- 5: **while** termination criteria is not met **do**
- 6: For each agent $\mathbf{s}_i \in P$ ($i = 1, 2, \dots, N$), randomly draw its d components $s_i^1, s_i^2, \dots, s_i^d$, from the interval $(\mathbf{x} - A, \mathbf{x} + A)$ by an uniform distribution. Also, by using Eq. 1, map each generated solution \mathbf{s} to be binary, and evaluate its objective function $f(\mathbf{s})$. Here, the objective function f calculates a number of covered components by the solution \mathbf{s} .
- 7: According to the Deb's rules, among all generated solutions \mathbf{s}_i ($i = 1, 2, \dots, N$), determine the smallest objective value as f_{min} . If $f_{min} < f(\mathbf{x})$ then update the current best solution \mathbf{x} with $\mathbf{x} \in \mathbf{s}_{min}$ and set $A \in C_a \cdot A$. Otherwise, keep the old best solution \mathbf{x} as the best one and set $A \in C_r \cdot A$.
- 8: **end while**
- 9: Return the best found solution \mathbf{x} .
-

Since an optimal camera placement is a combinatorial problem, and the original bare bones fireworks algorithm works with continuous problems, within the BBFWA, after the n solutions (sparks) \mathbf{s}_i ($i = 1, 2, \dots, n$) are initialized, each component s_i^j of the solution $\mathbf{s}_i = (s_i^1, s_i^2, \dots, s_i^d)$ is updated according to the following function:

$$s_i^j = \begin{cases} 1, & \text{if } r < s_i^j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where r is a uniformly distributed random number from the interval $(0, 1)$. The lower bounders Lb and upper bounders Ub for this problem were set to zero and one, respectively. Now, our adjusted version of the BBFWA only manipulates with the binary vectors, where one means that a vertex is pick up, and zero tell us that a vertex is not selected. After we described the mechanism for optimal interior covering of an orthogonal polygon, its realization is carried out by the ABBFWA approach whose necessary steps were summarized in the pseudo-code of Algorithm 3.

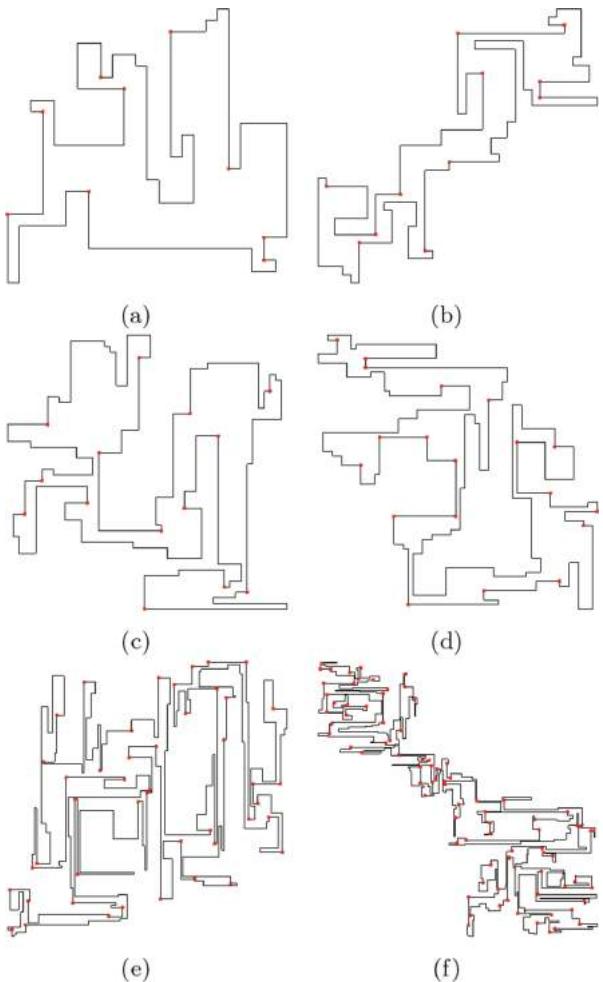


Fig. 2. From top to the bottom, the performance of our ABBFWA approach for randomly generated polygons: (a) RI-OP-50-4, (b) RI-OP-70-3, (c) RI-OP-100-2, (d) RI-OP-120-5, (e) RI-OP-250-4, (f) RI-OP-500-5

6 Simulation Experiment and Analysis of Results

In this simulation experimental, we have analysed results for our two proposed algorithms designed to cope with larger-scale problems. The purpose of the application our methods was to minimise the number of cameras required to perform interior coverage (IC) of the orthogonal polygons. The introduced techniques have been thoroughly examined to evaluate the essence of the results. The algorithms have been applied to 215 various randomly generated orthogonal polygons without holes which designed with our random orthogonal polygon generator developed for purposes of this paper similarly as in the paper [31].

Each instance is called RI-OP-k-i, where k denotes the size of the ith instance. The coordinates of points (x, y) are chosen from the interval $[0, 500]$. The offered methods have been realised in C# programming language. In order to estimate time efficiency and the coverage rate of the introduced algorithms, the comparisons of the obtained results have done by using a PC with an Intel Core i7-3770K @3.5 GHz with 64 GB of RAM running under the Windows 10 x64 operating system. As the ABBFWA approach is a swarm intelligence technique, in this paper, 200 agents and 100 iterations are allocated for its execution, which is a total of 20,000 functional executions. Also, the algorithm's parameters for search space size regulation, such as C_r and C_a were initialised on 0.7 and 1.2, respectively. Since our ABBFWA approach is a stochastic method, we ran it in parallel in 30 independent series using multi-threads, and we presented the obtained results in Table 1 and Table 2. In Table 1, to check the quality of obtained solutions as well as computational times, the algorithms were selected and tested through 7 groups of randomly generated orthogonal polygons, where each group was composed of five randomly generated polygons containing 20, 50, 70, 100, 120, 250 and 500 vertices, respectively. The coverage results of the ABBFWA approach for the orthogonal polygons whose instances are shown in Table 1, and which are composed of 50 vertices (the instance RI-OP-50-4), 70 vertices (the instance RI-OP-70-3), 100 vertices (the instance RI-OP-100-2), 120 vertices (the instance RI-OP-120-5), 250 vertices (the instance RI-OP-250-4) and 500 vertices (the instance RI-OP-500-5) are plotted in Fig. 2. The instances, as mentioned above, are extracted from Table 1 and shown in Fig. 2, since the ABBFWA approach requires the minimal number of cameras on them compared to the deterministic visibility algorithm. Also, the red dots in Fig. 2 indicate the locations of the cameras covering the whole interior of the polygons.

In Table 1, the abbreviations such as (Min.), (Avg.) and (Max.) indicate the best results, mean results and the worst results obtained after 30 independent series of execution of our ABBFWA approach, respectively. Besides, the statistical parameters, such as the standard deviation (Std.) and mean time (Mean time), talk about the stability and efficiency of the ABBFWA approach. On the other hand, as our second visibility algorithm is deterministic nature and it always gives the same results, it is driven once, and its results have recorded through the parameters such as No. of cameras and Time. From the obtained simulation results shown in Table 1, we can see that the ABBFWA in order to cover all orthogonal polygons from all seven groups requires 2010 cameras in the best case (Min.), 208.48 cameras in the average case (Avg.), and 2160 in the worst case (Max.). Opposite to the ABBFWA, our second visibility deterministic approach allocates 2182 cameras, which is more than 172 cameras compared to the smallest number of cameras determined by the ABBFWA approach. Even in the worst case, our first stochastic ABBFWA technique needs 22 fewer cameras than the second deterministic method. Particularly superiority comes to the fore with an increase in the size of polygon. In reality, reducing the number of cameras has many implications, and some of them are less money, energy savings, a less number of hardware components, a less cost of modifications on the network,

Table 1. The simulation results provided by our algorithms for 35 randomly distributed instances of the orthogonal polygons.

Random instances	ABBFWA approach					Visibility algorithm	
	Number of cameras			Std. dev.	Mean time (sec.)	No. of cameras	Time (sec.)
	Min	Avg	Max				
RI-OP-20-1	4	4	4	0	0.108	5	0.003
RI-OP-20-2	3	3	3	0	0.047	4	0.008
RI-OP-20-3	4	4	4	0	0.06	5	0.002
RI-OP-20-4	4	4	4	0	0.056	5	0.002
RI-OP-20-5	4	4	4	0	0.024	5	0.003
RI-OP-50-1	8	8.31	9	0.46263	0.086	10	0.041
RI-OP-50-2	8	8.48	10	0.56345	0.195	10	0.014
RI-OP-50-3	7	7.43	8	0.49851	0.246	9	0.022
RI-OP-50-4	9	9	9	0	0.195	12	0.02
RI-OP-50-5	9	9.69	10	0.46263	0.19	10	0.019
RI-OP-70-1	10	10.59	13	0.67042	0.31	12	0.031
RI-OP-70-2	13	13.07	14	0.2534	0.371	15	0.033
RI-OP-70-3	11	11.31	12	0.46263	0.312	14	0.031
RI-OP-70-4	11	11.52	13	0.56451	0.331	13	0.029
RI-OP-70-5	13	13.07	14	0.2534	0.371	15	0.033
RI-OP-100-1	17	18	19	0.69481	0.594	20	0.06
RI-OP-100-2	14	14.69	16	0.69992	0.543	18	0.072
RI-OP-100-3	16	16.38	17	0.49372	0.967	19	0.068
RI-OP-100-4	17	17.96	20	0.99941	0.604	20	0.069
RI-OP-100-5	15	15.21	16	0.37931	0.553	18	0.063
RI-OP-120-1	20	20.72	22	0.58111	0.765	23	0.107
RI-OP-120-2	22	22.62	24	0.66508	0.81	25	0.089
RI-OP-120-3	20	20.83	22	0.73068	0.795	23	0.092
RI-OP-120-4	19	20.1	21	0.60713	0.778	23	0.085
RI-OP-120-5	19	20.45	22	0.8919	0.777	23	0.096
RI-OP-250-1	41	41.43	45	0.96552	2.593	44	0.562
RI-OP-250-2	43	44.9	47	1.04193	2.69	46	0.597
RI-OP-250-3	39	42.65	45	1.25235	1.792	43	0.585
RI-OP-250-4	45	48.03	50	1.2452	2.591	50	0.548
RI-OP-250-5	45	46.96	49	0.96675	2.664	49	0.475
RI-OP-500-1	90	93.27	97	1.67481	10.498	95	3.143
RI-OP-500-2	84	87.55	91	1.92147	9.415	89	2.501
RI-OP-500-3	88	90.72	96	1.56356	9.38	96	2.436
RI-OP-500-4	82	85.96	89	1.78912	10.1	91	2.894
RI-OP-500-5	84	87.45	91	1.64905	9.762	93	2.413
RI-OP-1250-1	211	215.83	220	2.64845	104.878	220	31.223
RI-OP-1250-2	218	224.51	232	2.88421	106.871	230	40.014
RI-OP-1250-3	221	226.72	231	2.54776	94.772	232	30.342
RI-OP-1250-4	213	220.76	227	2.80203	100.21	223	32.957
RI-OP-1250-5	209	215.31	220	2.7502	89.345	225	29.607

Table 2. The average number of cameras and meantime processing provided by our algorithms for 210 randomly generated instances.

		ABBFWA approach			Visibility algorithm		
No. rand. instances	Size (n)	Mean no. of cameras			Mean time (s)	Mean no. cameras	Mean time (s)
		Best	Average	Worst			
30	20	3.47	3.47	3.47	0.06	3.83	0.000
30	50	8.32	8.49	9.14	0.20	9.23	0.020
30	70	11.50	11.87	12.70	0.34	12.40	0.030
30	100	16.47	17.07	18.30	0.60	17.83	0.070
30	120	20.20	20.96	22.40	0.80	21.93	0.100
30	250	41.80	43.78	46.23	2.48	44.47	0.422
30	500	85.00	88.32	92.10	9.97	90.07	2.770

and so forth. Based on the results shown in Table 1, we can note that ABBFWA provides the highest quality solutions. On the other hand, our second suboptimal deterministic method is usually get trapped in some local optima, and as a consequence of it does not able to find the global optimum. In order to show the real robustness of the proposed methods, we tested them for a dataset composed of 210 randomly generated orthogonal polygons, and the results obtained were saved in Table 2. The simulation results show that the ABBFWA gets in any case, better quality solutions compared with the approximate one for all sizes of the polygons. Other words, the mean number of cameras increases linearly concerning the size of vertices (n) so that a growth rate of the cameras being noticeably slower in the ABBFWA method compared to the price of growth generated by the suboptimal deterministic approach. Also, by considering produced experimental results in Table 2, we can conclude that all methods are comparable in terms of CPU execution time. Also, from results shown in Table 2, we can conclude that on the best case the minimum number of cameras needed to perform IC of an orthogonal polygon P with n vertices is less or equal $\lfloor \frac{n}{6} \rfloor$, which is a much better estimate than $\lfloor \frac{n}{4} \rfloor$ proposed by Lee and Lin [20]. Based on the experimental analysis, it can be concluded that metaheuristics are an appropriate practical tool that is capable of handling camera placement problem, which has direct applications in the area of intelligent video surveillance which can automatically identify potential risks by detecting, localizing, tracking and recognizing targets or events of interest.

7 Conclusion and Future Work

Camera networks are complex systems capable of receiving broad video information for intelligent processing jobs, such as target localization, identification, and tracking. In this article, we investigated the optimal camera placement problem

for interior coverage of the simple orthogonal polygons and proposed two versions of algorithms for its solving. Quality of our methods was tested throughout 215 randomly generated instances. Based on the experiment research, it can be inferred that our first ABBFWA approach is convenient for this task, and it produces excellent overall performance. Also, the ABBFWA approach proved to be robust, in the sense that it was able to tackle different instances from a broad range of randomly generated ones. Since the computational preprocessing time of the ABBFWA is computationally expensive for large instances, in future work, we will investigate the efficient techniques from computational geometry in order to tackle these drawbacks. Also, further research will focus the orthogonal polygons with holes as well as on how the proposed ABBFWA approach can be applied to other three types of real cameras, such as static perspective camera, pan-tilt-zoom (PTZ) camera, an omnidirectional camera.

References

- Alihodzic, A.: Fireworks algorithm with new feasibility-rules in solving UAV path planning. In: 2016 3rd International Conference on Soft Computing Machine Intelligence (ISCMI), pp. 53–57 (2016)
- Altahir, A.A., et al.: Optimizing visual surveillance sensor coverage using dynamic programming. *IEEE Sens. J.* **17**(11), 3398–3405 (2017)
- Bhuiyan, M.Z.A., Wang, G., Cao, J., Wu, J.: Deploying wireless sensor networks with fault-tolerance for structural health monitoring. *IEEE Trans. Comput.* **64**(2), 382–395 (2015)
- Bhuiyan, M.Z.A., Wang, G., Cao, J., Wu, J.: Sensor placement with multiple objectives for structural health monitoring. *ACM Trans. Sen. Netw.* **10**(4), 1–45 (2014)
- Bjorling-Sachs, I., Souvaine, D.L.: An efficient algorithm for guard placement in polygons with holes. *Discrete Comput. Geomet.* **13**, 77–109 (1995)
- Bodor, R., Drenner, A., Schrater, P., Papanikopoulos, N.: Optimal camera placement for automated surveillance tasks. *J. Intell. Rob. Syst.* **50**(3), 257–295 (2007)
- Chrysostomou, D., Gasteratos, A.: Optimum multi-camera arrangement using a bee colony algorithm. In: 2012 IEEE International Conference on Imaging Systems and Techniques Proceedings, pp. 387–392 (2012)
- Chvátal, V.: A combinatorial theorem in plane geometry. *J. Combin. Theory Ser. B* **18**(1), 39–41 (1975)
- de Berg, M., Cheong, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications, 3rd edn. Springer, Heidelberg (2008). <https://doi.org/10.1007/978-3-540-77974-2>
- Elnagar, A., Lulu, L.: An art gallery-based approach to autonomous robot motion planning in global environments. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 2079–2084 (2005)
- Feng, G., Liu, M., Guo, X., Zhang, J., Wang, G.: Genetic algorithm based optimal placement of pir sensor arrays for human localization. In: 2011 IEEE International Conference on Mechatronics and Automation, pp. 1080–1084 (2011)
- Fisk, S.: A short proof of Chvátal's watchman theorem. *J. Combin. Theory Ser. B* **24**(3), 374 (1978)
- Gonzalez-Barbosa, J., Garcia-Ramirez, T., Salas, J., Hurtado-Ramos, J., Rico-Jimenez, J.: Optimal camera placement for total coverage. In: 2009 IEEE International Conference on Robotics and Automation, pp. 844–848 (2009)

14. Györi, E., Hoffmann, F., Kriegel, K., Shermer, T.: Generalized guarding and partitioning for rectilinear polygons. *Comput. Geom.* **6**(1), 21–44 (1996)
15. Hoffmann, F., Kaufmann, M., Kriegel, K.: The art gallery theorem for polygons with holes. In: 1991 Proceedings 32nd Annual Symposium of Foundations of Computer Science, pp. 39–48, October 1991
16. Islam, S.H., Vijayakumar, P., Bhuiyan, M.Z.A., Amin, R., Rajeev M.V., Balusamy, B.: A provably secure three-factor session initiation protocol for multimedia big data communications. *IEEE Internet Things J.* **5**(5), 3408–3418 (2018)
17. Kahn, J., Klawe, M., Kleitman, D.: Traditional galleries require fewer watchmen. *SIAM J. Algebraic Discrete Methods* **4**(2), 194–206 (1983)
18. Kamkar, S., Ghezloo, F., Moghaddam, H.A., Borji, A., Lashgari, R.: Multiple-target tracking in human and machine vision. *PLOS Comput. Biol.* **16**(4), 1–28 (2020)
19. Katz, M.J., Roisman, G.S.: On guarding the vertices of rectilinear domains. *Comput. Geom.* **39**(3), 219–228 (2008)
20. Lee, D., Lin, A.: Computational complexity of art gallery problems. *IEEE Trans. Inf. Theory* **32**(2), 276–282 (1986)
21. Li, J., Zheng, S., Tan, Y.: Adaptive fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3214–3221 (2014)
22. Li, J., Zheng, S., Tan, Y.: The effect of information utilization: introducing a novel guiding spark in the fireworks algorithm. *IEEE Trans. Evol. Comput.* **21**(1), 153–166 (2017)
23. Li, J., Tan, Y.: The bare bones fireworks algorithm: a minimalist global optimizer. *Appl. Soft Comput.* **62**, 454–462 (2018)
24. Liu, J., Fookes, C., Wark, T., Sridharan, S.: On the statistical determination of optimal camera configurations in large scale surveillance networks. In: Computer Vision - ECCV 2012, pp. 44–57. Springer, Heidelberg (2012)
25. Liu, J., Sridharan, S., Fookes, C.: Recent advances in camera planning for large area surveillance: a comprehensive review. *ACM Comput. Surv.* **49**(1), 1–37 (2016)
26. O'Rourke, J.: Art Gallery Theorems and Algorithms. Oxford University Press, Oxford (1987)
27. O'Rourke, J.: Computational Geometry in C. Cambridge University Press, Cambridge (1998)
28. O'Rourke, J., Supowit, K.: Some np-hard polygon decomposition problems. *IEEE Trans. Inf. Theory* **29**(2), 181–190 (1983)
29. Schuchardt, D., Hecker, H.-D.: Two np-hard art-gallery problems for ortho-polygons. *Math. Log. Q.* **41**(2), 261–267 (1995)
30. Tan, Y., Zhu, Y.: Fireworks algorithm for optimization. In: Advances in Swarm Intelligence. LNCS, vol. 6145, pp. 355–364 (2010)
31. Tomás, A.P., Bajuelos, A.L.: Generating random orthogonal polygons. In: Conejo, R., Urretavizcaya, M., Pérez-de-la Cruz, J.-L. (eds.) Current Topics in Artificial Intelligence, LNCS. vol. 3040, pp. 364–373. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-25945-9_36
32. Tuba, E., Hrosik, R.C., Alihodzic, A., Jovanovic, R., Tuba, M.: Support vector machine optimized by fireworks algorithm for handwritten digit recognition. In: Simian, D., Stoica, L.F. (eds.) Modelling and Development of Intelligent Systems, vol. 1126, pp. 187–199. Springer, Heidelberg (2020). https://doi.org/10.1007/978-3-030-39237-6_13
33. Tuba, M., Bacanin, N., Alihodzic, A.: Multilevel image thresholding by fireworks algorithm. In: 2015 25th International Conference Radioelektronika (RADIOELEKTRONIKA), pp. 326–330 (2015)

34. Wang, T., et al.: Energy-efficient relay tracking with multiple mobile camera sensors. *Comput. Netw.* **133**, 130–140 (2018)
35. Yao, Y., Chen, C., Abidi, B., Page, D., Koschan, A., Abidi, M.: Can you see me now? Sensor positioning for automated and persistent surveillance. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **40**(1), 101–115 (2010)
36. Yu, C., Tan, Y.: Fireworks algorithm with covariance mutation. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 1250–1256 (2015)
37. Zhao, J., Cheung, S., Nguyen, T.: Optimal camera network configurations for visual tagging. *IEEE J. Sel. Top. Signal Process.* **2**(4), 464–479 (2008)
38. Zheng, S., Janecek, A., Li, J., Tan, Y.: Dynamic search in fireworks algorithm. In: 2014 IEEE Congress on Evolutionary Computation (CEC), pp. 3222–3229 (2014)
39. Zheng, S., Li, J., Janecek, A., Tan, Y.: A cooperative framework for fireworks algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **14**(1), 27–41 (2017)



Unsupervised Machine Learning-Based Elephant and Mice Flow Identification

Muna Al-Saadi^(✉), Asiya Khan, Vasilios Kelefouras, David J. Walker,
and Bushra Al-Saadi

University of Plymouth, Plymouth, UK

{muna.al-saadi, asiya.khan, vasilios.kelefouras, david.walker,
bushra.al-saadi}@plymouth.ac.uk

Abstract. Internet today holds traffic from a wide range of applications, which have different requirements and constraints on the resources of a network. Hence, it is normal to find a variety of flows with dissimilar features that contend for the network resources. Consequently, the problem that appears clearly is an unfair use of these resources by particular flows. This problem exposed to the so-called elephant and mice flows through real analysis of network traffic. Therefore, this problem might lead to degrading network performance. In this paper, we proposed a framework to optimize the network performance through characterising elephant and mice flows based on network performance metrics. The framework has three parts. Principal component analysis (PCA) is used in the first part to reduce the dimensionality. The next part was responsible for partitioning the traffic into distinct groups based on performance metrics such as packet loss, round trip time (RTT), and throughput by using an unsupervised clustering method with k-means. Finally, for each cluster, flows have been identified as huge (elephant) and small (mice) based on threshold values for the predefined parameters. Our results show that there is a potential in using network performance features to cluster the network traffic and to identify mice and elephant flows based on the number of packets, flow size, and duration of flow. We analyzed a (2 GB pcap file) to build our dataset. Finally, our proposed framework is capable of characterizing mice and elephant flows based on network performance metrics for each cluster.

Keywords: Machine learning · Network performance · PCA · k-Means · Elephant and mice flows

1 Introduction

The main objective of network management is to preserve the availability of a network and improve its performance. Network management is becoming clearly challenging with the growth of network size, traffic volume, and the diversity of requirements in Quality of Service (QoS). The Internet today carries traffic from a wide range of applications, which have different requirements and constraints on the resources of the network. Hence, it is normal to find diverse flows with dissimilar features that compete

for network resources. Therefore, the network resources are not equally used by all the flows.

This problem can be tackled with the so-called elephant and mice flows through real analysis of the network traffic. For example, elephant flows can fill network buffers end to end, which can lead to queuing delays, impacting latency-sensitive flows, also known as mice. Therefore, this problem might lead to poor network performance. Recent studies in network engineering propose new strategies to optimize network performance by identifying and handling mice and elephant flows differently. These studies include the assignment of different flows to different queues [1], flow distribution across the links [2], and creating a policy of routing as a rule [3]. Identifying and differently treating mice and elephant flows accomplished by managing priority, or rerouting is required to guarantee the performance improvement of the network.

Nowadays, each service provided by the network has different requirements for its continuity. To identify these requirements a precise understanding of network behaviour is required. Machine learning (ML) has been employed for this purpose. ML has been successfully used to extract knowledge from data in a range of domains where a wide variety of its methods were used to classify network traffic, using statistical analysis based on conceptual classification. Various statistical features related to the packets of flow such as the size of the packet, number of packets, inter-arrival time, duration, and are used by ML for classification purposes.

Network researchers have proposed methods to provide good QoS to flows by prioritizing mice flows or rerouting elephant flows. These methods are achieved either by applying machine learning approaches, used certain data structures, design adaptive routing architecture, or proposed a flow scheduling algorithm.

The aforementioned studies have used one or two variables, such as the size of flow and the duration, to distinguish between elephant and mice flows. In addition, they have assigned different threshold values for the identification process, 100 KB to 10 MB for total byte count per flow and 10s for duration threshold. Due to the ever increasing Internet traffic, improving the network flow is still a challenge and of considerable interest to network operators. It is now well established that the identification of elephant and mice flows has a significant role in the enhancement of network performance. However, the influence of the characterizing of elephant and mice flows on the identification process has remained unclear.

Hence, in this paper we apply machine learning techniques of unsupervised learning to cluster the network traffic based on features. We captured real data with 2GB pcap file. We first apply principal component analysis to reduce the dimensionality of the data while retaining the features. We then used k-means to cluster the data based on similar attributes to help flow identification of elephant and mice. This has been achieved based on a specific thresholds and features. Therefore, the main contribution of this paper is two-fold:

1. to use reduce the dimensionality of the data and apply unsupervised machine learning technique of k-means
2. based on (i) above, to introduce a new identification mechanism for mice and elephant flow identification

The rest of the paper is organized as follows. Section 2 presents related works. The proposed flow identification methodology is presented in Sect. 3. Section 4 contains the experimental set-up. Results are discussed in Sect. 5, and Sect. 6 concludes the work.

2 Related Work

Machine Learning (ML) is the term given to a set of powerful techniques for data extraction and the discovery of knowledge. Furthermore, one of the most favorable techniques to carry out network-data analysis and to automate configuration and management of networks because of its ability to make network elements 'learn' from experience by using the large quantity of data to make networks more intelligent and adaptive[4]. k-Means is a clustering algorithm, which stores particular pre-chosen k centres which it utilises to generate clusters randomly, according to the similarity (often Euclidean distance) between all input objects [5].

There is a wide range of clustering applications in networking. Clustering is effective in a range of fields, including network management and security. In [6], similarity-based clustering of application traffic was executed by using two unsupervised clustering approaches: k-means and Expectation Maximization (EM). In addition, the Correlation-based Feature Selection (CFS) filter method was used to select appropriate attributes of a flow. According to researchers in [7] concentrate on using ML methods for statistical flow-based traffic classification. The authors propose a new framework, which is Traffic Classification using Correlation (TCC) information, to handle the problem of very few training samples. The framework has constructed by selecting suitable features and the correlated information of TCP flows to enhance the classification performance. In [8] a new unsupervised method for traffic classification was suggested to solve the problem of unknown applications through identifying traffic classes, based on flow statistical features, where automated flow classification and signature-based cluster aggregation were executed by finding a similarity between traffic clusters. While in [9], the C5.0 technique was used for application classification. A new set of features, which are burstiness and idle time, have proposed to determine the type of applications that generate the traffic. The proposed features have proved their effectiveness to identify the type of applications as compared with previous studies. Another study of the use of unsupervised ML algorithms to identify applications was described in [10]. In this study, seven different application groups were concentrated. This work introduces a system, executed by a set of tasks that measures the network's QoS. In accordance with [11], an amended k-means-based semi-supervised clustering method was used. Flow statistics-based traffic classification was applied, where layer four statistics were considered as inputs, to manipulate the classification process.

Two difficulties with modeling are the curse of dimensionality and overfitting especially when we have many features and relatively few samples. One popular approach is dimensionality reduction. Principal Component Analysis (PCA) is a commonly used method for this purpose. The obtained results in [12] and [13] were visualized as 3D by reducing the dimensions of the dataset and to calculate the correlation values between each feature and each of the 13 principal components that have been chosen using PCA. Study in [14] proposes a new technique of routing by integrating unsupervised machine

learning and Software Defined Network (SDN) environment, where PCA has been used for dimensional reduction, the k-means algorithm to group the identical flows based on performance metrics, which will be implemented within SDN paradigm. Although the accuracy of k-means clustering with PCA approximates to 100%, the dataset used was small (11593 connections). In addition, the number of PCA components (5 components) that have been chosen, covered a small percentage of data (40%).

On the other hand, the problem of precise identification of mice and elephant flows is still needs to be handled. In [15], the authors proposed a method for identifying flows as mice or elephants. Unsupervised and semi-supervised ML approaches have been used for classification flows in real-time. They used three parameters as data transfers, flow rates, and durations for clustering. The accuracy of the proposed method was 90%. While the authors in [16] proposed a system that detects elephant flows in linear time through traffic volume estimation for every flow. The elephant flows identified by counting their total bytes. They used a certain data structure (hash-tables) to achieve elephant identification. This system applied two algorithms for maintaining the data structure, Median SUMming (IM-SUM) and De-amortized Iterative Median SUMming (DIM-SUM), where they use two different tables for calculation. On the other hand, other studies clarified that identification and rerouting the flows that hold a large amount of data (elephant flows) effectively can lead to significant improvement in QoS. Authors in [17] presented a system called DARD (Distributed Adaptive Routing architecture for Datacentre networks). This system consists of three parts. The first is for detecting an elephant flow if a TCP connection has lasted for more than 10s. The second is a tracking monitor to check if all the paths connecting the source and destination switches are existing. The third part is a flow scheduler that shifts elephant flows from overloaded paths to under-loaded ones. Another approach is proposed by [18], authors present a flow scheduling algorithm, which dynamically adjusts the number of two types of paths according to the real-time traffic into low latency paths and high throughput paths respectively for the two types of flows to make full utilization of the bandwidth. As proposed in [19], a routing algorithm called Distributed Flow Scheduling (DiFS) system, defined the flow, which exceeds a threshold of 100 KB as a large flow. After that, this flow will be transmitted to the destination by switching to a path with an abundant amount of bandwidth.

Recent works have been applying machine learning in network traffic classification. However, the limitations of existing studies are in the number of connections used as well as the features identified. The novelty in our work is that we have first reduced the dimensionality of the data by applying PCA with different numbers of components (124, 57, 28, and 13 respectively) for data reduction. Further, k-means were applied to cluster the traffic based on network performance metrics using a dataset that contained 1 million complete connections. Finally, full and precise identification of flows as elephant and mice was performed using pre-decided threshold values.

3 Proposed Methodology

This work proposes a method that addresses the identification problem of elephant and mice flows and characterising them by leveraging network performance metrics such as packet loss, round trip time (RTT), and throughput. Figure 1 shows the main stages of the

proposed method, which include reducing the dimensionality of the dataset, clustering network traffic using unsupervised ML, and identifying elephant and mice flows based on predefined parameters for each cluster. The detailed steps of the proposed method are discussed hereafter:

First, data pre-processing applied to facilitate the ML step. This part of the work includes manipulating the raw data, which contains missing values and non-numeric values in order to produce a dataset of TCP parameters dataset to be used as input for the method. The tcptrace analysis tool [20] can provide a comprehensive set of features including 146 attributes associated with the network connection. The data-cleaning step used to clean the data set by removing irrelevant and useless records and features. All features with character values converted to numeric values. The resultant dataset is normalized as the final step of this part.

The next step was to apply the Principal Component Analysis (PCA) technique to reduce dimensionality. PCA is a linear transformation technique, which transforms data such that the data with the highest variance represented as the first coordinate, and the data with second-highest variance put in the second coordinate, and so on. Therefore, by keeping the coordinates with high variance values and ignoring the data that has low variance, the large dimension of datasets will be reduced. In this work, PCA executed with different number of components to cover different percentages of data to show the effects of the number of components on the accuracy of clustering.

Then, an unsupervised clustering algorithm used after PCA to cluster the traffic flows, based on their performance parameters. The unsupervised machine learning works with unlabeled data. The clustering aims to build a robust cluster labeling. After running k-means on the initial data that is not the end of the lifecycle of the proposed model. Therefore, we can interpret new data that contain unseen points as test data. That presupposes that we have trained the model on our data, which would then be considered training data. This technique implemented for incremental values of k to see how the

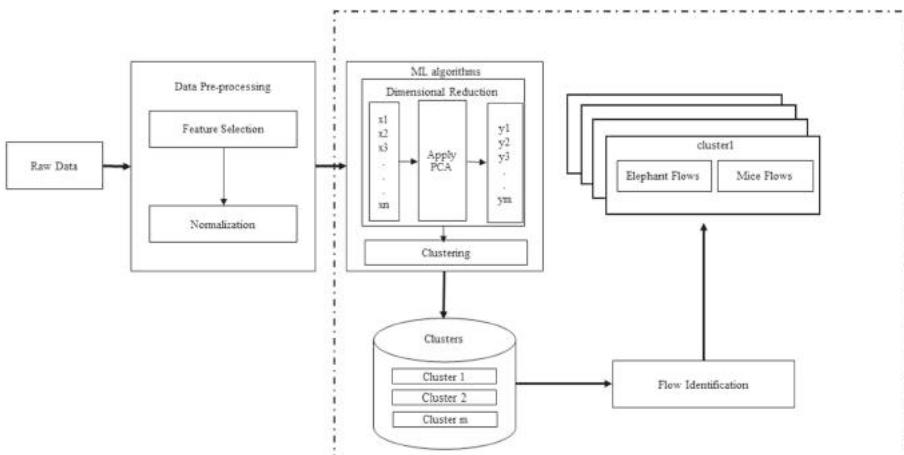


Fig. 1. Proposed methodology

accuracy of clustering has changed with changing the number of PCA components and the number of clusters (k).

Finally, after the clustering process, for each resultant cluster, mice and elephant flows identified. Classification of flows is either done via human experience or using thresholds based on pre-defined parameters, to identify long-lived versus short-lived flow. In this step, specific threshold values for pre-set parameters (number of packets, flow size, and flow duration, respectively) used to identify elephant and mice flows. These parameters will be outlined shortly. The identification process was efficient because it required less processing time and improved the accuracy of results.

4 Experimental Set-Up

This section describes the dataset, the process of building the clusters and the experimental results. Firstly, we determine the criteria that will be used to choose the connections, the features that will be extracted, and the steps of pre-processing process that will be needed for this dataset, and the rationale behind it. Finally, machine learning methods will be used. The R scripting language has been used in the experiment.

4.1 Data Description

Our model has been run experimentally on real-world data captured as raw data through one hour in a laboratory stored in a pcap file. The size of the file was 2GB. The tcptrace tool used to analyze the raw traffic data and converted it to a dataset containing 2 million connections. It is noteworthy that complete connections have been selected for the entire image of network performance. For that reason, a dataset of 1 million complete connections was used. A connection is constructed from a sequence of TCP packets on two directions. A vector of 146 extracted attribute values has represented each connection. These attributes are associated with the network performance such as duration, the number of bytes, packets transferred, round trip time, the number of retransmissions, window advertisements, and throughput. The resultant dataset requires a pre-processed step to handle incomplete, noisy, and inconsistent data points as described in the next section.

4.2 Data Pre-processing

To build a highly accurate dataset, pre-processing steps have applied. These steps started by eliminating all missing values or inconsistent data from the dataset. Then encoding of categorical data was done by converting it to numerical values such as control features (FIN and SYN features) have performed. The final step is data normalization or scaling, which is a method to standardize the variables of a dataset within a specific range to ensure that the unit of feature keeps the closeness of cases. To achieve this step, for each value of a variable, we simply subtract the mean value of the variable, and then divide it by the standard deviation of it. As a result, a dataset contains 129 features and 1 million complete connections will be the input to the machine learning model.

4.3 Data Reduction

PCA is one of the common approaches that has been used recently to evolve a low dimensional set (components format) of variables from high dimensional data set while keeping as much information as possible. Because our dataset has more than one hundred numeric variables, which characterize each connection, this technique has been used in this work. Furthermore, PCA is undertaken in cases when there is sufficient correlation among the original variables, which is the second reason to use PCA in this work where some of our variables have significant correlation, to warrant component representation. Because our dataset contains extreme values that are outside the range of what is expected and unlike the other data, these are the outliers that can negatively affect the clustering process. To handle the outliers, PCA has been used. As a general rule, the number of components will be equal to the number of variables in the dataset. All resultant components explain the full variation in data. Since PCA is a dimension reduction method, therefore there is a need to retain a suitable number of components based on the trade-off between completeness and simplicity. In this work, the “quick.elbow” function has been used to choose a suitable number of components. This function decides how many components should be retained based on their cumulative percentage. We have run PCA with a different number of components to decide the suitable number of components to achieve good clustering (clustering with high accuracy).

4.4 K-mean Clustering

The k-means is considered one of the simplest and fastest clustering method. The role played nowadays by classification techniques in a variety of network fields, such as network management and security, is very effective. Therefore, the k-means algorithm was used in this section of work. The criteria used to execute k-means are the number of clusters (k) and the number of maximum iterations. In this part, the k-means algorithm has run for incremental values of k from 10 to 100 in steps of 10, where the cluster centers initialized randomly. The metric that has been used to measure the goodness of clustering without comparing the results of cluster analysis to external information (ground truth or class labels) is represented by the accuracy. The accuracy depends on inter-cluster distance and intra-cluster distance.

4.5 Flow Labelling

In networking, a flow is defined as a sequence of packets that can be described by Source IP, Source Port, Destination IP, Destination Port, Protocol (TCP, UDP, etc.). Mice flows are identified as small volume, short-lived flow. In contrast, elephant flows are large, long-lived flows. On the Internet, most of the flows, which comprise 90% of all flows, are short and transmit a small amount of traffic (mice), while elephant flows constitute only 10% of all flows, but transmit a large amount of bytes.

The automated identification and specification of mice flow as a prior over elephant flows can improve transaction computation, which is based on a small data block, and optimize the network forwarding of these flows. These can contribute with a 30% reduction in the completion time of the application [21]. To identify mice and elephants, the

choice of the appropriate threshold can be flexible. Cisco's ACI identifies small flows as anything less than 15 packets. On the other hand, in [10], the size of mice flows is typically less than 10 KB and therefore it is just a few packets. In [11], it is rare that the mice flows exceed 5 s, therefore most the long-duration flows are the elephant.

In this work, the previous features of flow (number of packets, flow size, and duration of flow) and their values (15 packets, 10 KB, and 5 s) respectively, will be used to identify flows in each cluster.

5 Results

This section presents the experimental results of Principal Component Analysis (PCA), k-means clustering and the identification of mice and elephant flows for each cluster.

5.1 Principal Component Analysis

PCA has been used as a first phase to analyse the correlation between the features for the dimensional reduction in order to make the clustering algorithm more effective and efficient.

The dataset of this work has over one hundred variables to describe each connection. Furthermore, there is a significant correlation between these variables. Figure 2 depicts the correlation of the first ten variables in a dataset. In this figure, the correlation coefficient has been coloured according to the value. Blue circles represent positive

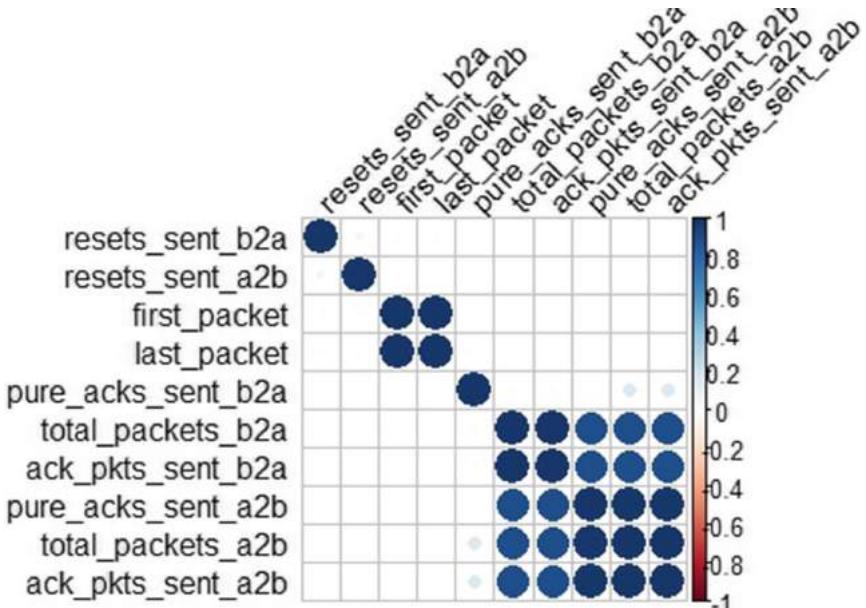


Fig. 2. Correlation of the first ten network performance variables.

correlation, while red circles display negative correlations. The size and color intensity of the circles is proportional to the correlation coefficients.

For the dataset with 192 performance features, the proportion of variance explained by each principal component was shown in Fig. 3. Visualizing the variance explained by each component helps us in identifying how many principal components are needed to explain the variation in data.

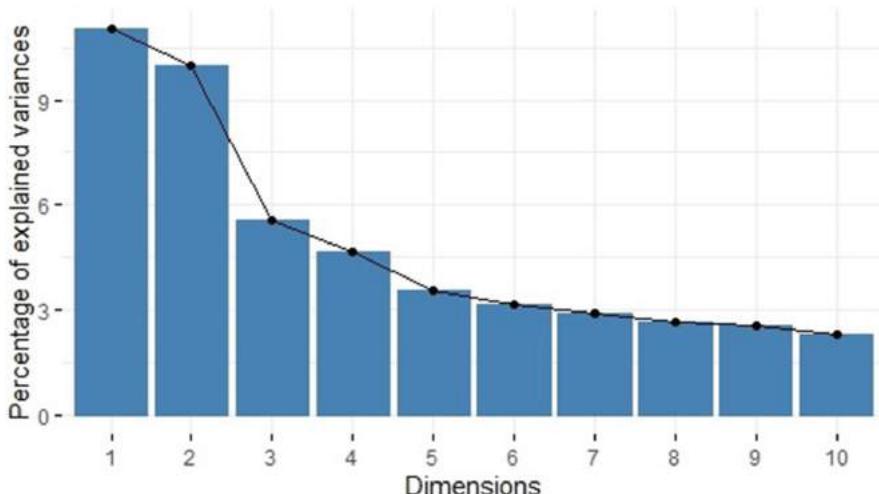


Fig. 3. Percentage of variance in each principle component.

We notice that the first three principal components explain most of the variation in our data. As these components represent less than 30% of the data we need to increase the number of components to represent more data. Table 1 depicts the first 13 components and their associated eigenvalues, the proportion of variance, and cumulative variance. It is significant to retain a suitable number of components based on the trade-off between completeness and simplicity. In this work, we experiment with different numbers of PCA components (124, 57, 28, and 13) which represent 100%, 90%, 70%, and 50% of data respectively. For all 129 variables, about 27 features contributed the most variance to the overall variance.

Table 1. The First 13 PCA Components.

Component	Eigenvalues	Variance (%)	Cumulative (%)
Comp1	14.22	11.03	11.03
Comp2	12.84	9.95	20.98
Comp3	7.18	5.57	26.55
Comp4	6.03	4.68	31.23
Comp5	4.57	3.55	34.77
Comp6	4.10	3.17	37.95
Comp7	3.77	2.93	40.87
Comp8	3.41	2.64	43.52
Comp9	3.28	2.54	46.06
Comp10	2.97	2.31	48.36
Comp11	2.94	2.28	50.64
Comp12	2.87	2.22	52.87
Comp13	2.65	2.05	54.92

5.2 K-Means Clustering

We applied k-means clustering using both the original 129 features as input and the 13 principal components as input. k-means clustering has been performed using k from 10 to 100 and with different numbers of principal components (124, 57, 28, and 13, respectively). Since we have used unlabeled data in this experiment, the internal index, which is the accuracy, is used to measure the goodness of the clustering structure. The accuracy of clustering calculated is based on (i) inter-cluster distance between the observation and cluster centre, and (ii) intra-cluster distance between cluster centers. In general, the accuracy of clustering substantially increases as the number of clusters, k increases as Fig. 4 shows. In addition, the overall accuracy improves at least 10% after applying PCA with minimizing the number of components. The results show that using PCA before applying a clustering algorithm contributes significantly to increase the accuracy of clustering and minimizing the number of components. In contrast, the accuracy of clustering decreased when we take 124 components, which represent 100% of data as shown in Fig. 4. The 124 components that cover 100% of the data contain most of the features even the features with low variance. This will change the distribution of data among the clusters. As a result, the accuracy of clustering will be affected.

The main aim of this part of the experiment includes dimensional reduction and clustering process, and the criteria were to obtain high accuracy in the clustering process and obtain the balance between simplicity and completeness while retaining the appropriate number of principal components. From the results shown in Fig. 4, 13 principal components (representing 50% of data) and 60 clusters ($k = 60$) give high accuracy of 85.39%, are the best to be chosen. Figure 5 shows the distribution of data points in each cluster, where each color is a cluster identity for 60 clusters.

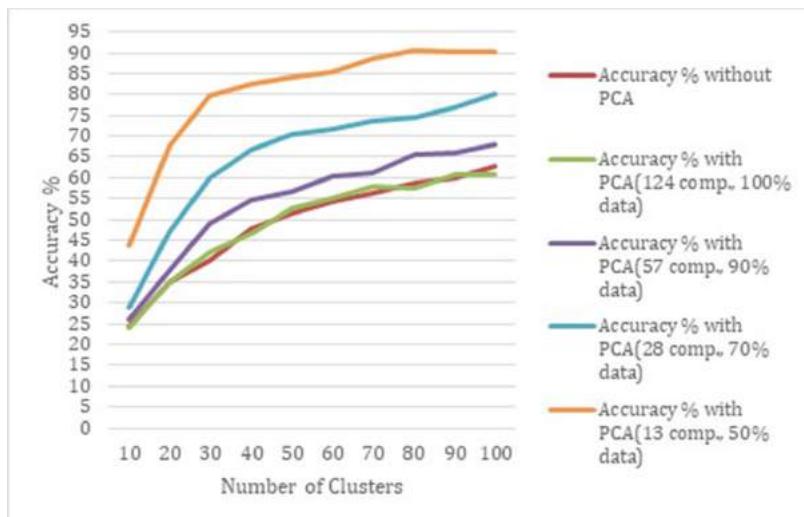


Fig. 4. Accuracy of clustering

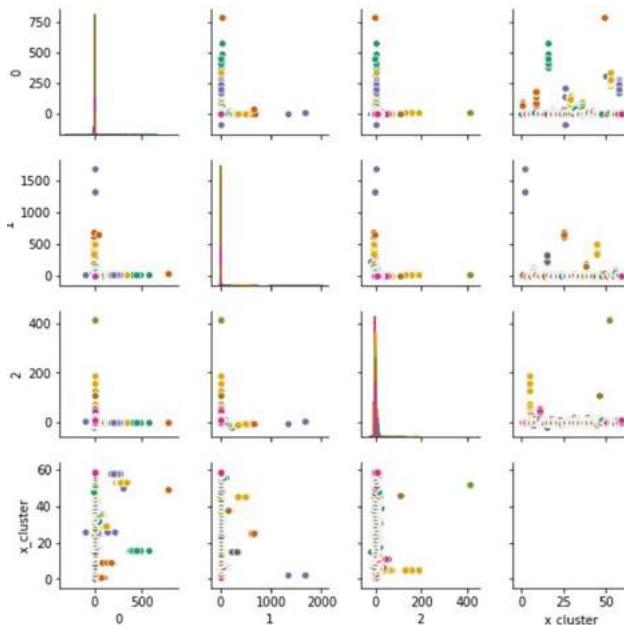


Fig. 5. Distribution of points in clusters

5.3 Flow Identification

This part of the experiment includes the identification of mice and elephant flows for each cluster overall 60 clusters based on a specified threshold (15, 10 KB, and 5 s) for

certain features of flow (number of packets, flow size, and duration of flow) respectively. The algorithm was implemented in the Python language. Figure 6 presents the identified elephant and mice flows in one of the 60 clusters. 89.92% represents the percentage of mice flows in the specified cluster, while the rate of elephant flows is 10.07%. This percentage is different from each cluster to the other cluster.

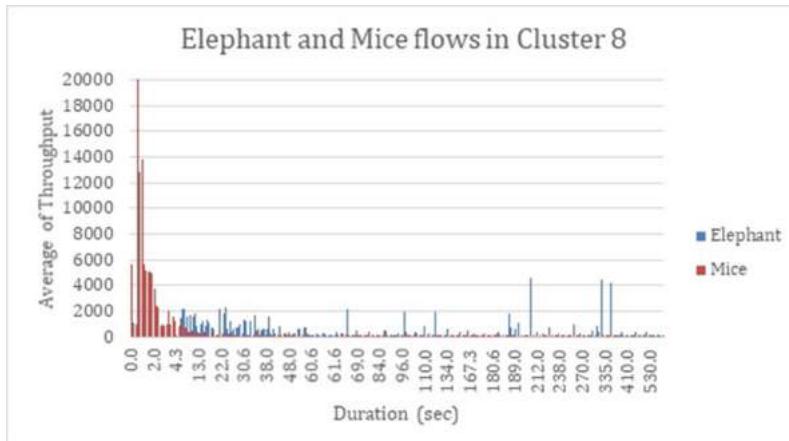


Fig. 6. Elephant and mice identification

5.4 Discussion

As mentioned in the literature review, the need to handle the precise identification of mice and elephant flow still exists. An initial objective of the project was to characterising network traffic based on network performance features. The current study found that unsupervised ML techniques could be utilised to cluster the network traffic based on the network performance metrics. However, the results showed that as the number of clusters (k) increases, the number of principal components has a substantial effect on the accuracy of the clustering process. Therefore, choosing an optimal number of components that achieved the balance between simplicity and completeness was crucial. These results corroborate the findings of a great deal of the previous work in [14]. For the second objective of this study that associated with elephant and mice flows identification, which is the main part of the experiment. The authors in [15] showed that initial findings with k-means proved that this technique has no ability to produce two distinct clusters because there is some information that related to the file sizes, which was difficult to group. This differs from the findings presented here where the thresholding approach has used to identify flows as elephant and mice based on pre-decided parameters. The findings showed that the selected thresholds were efficient for the identification process. One interesting finding in this experiment is that there is potential in using network performance features to characterise the mice and elephant flows. Our findings may be somewhat limited by it was not possible to completely avoid outliers that lead to

the presence of clusters with a size beyond the limits of clusters convergence. Further research should be undertaken to investigate the possibility of identifying elephant and mice flows using an SDN environment.

6 Conclusions

This paper has presented the use of network performance features to cluster the network traffic with the identification of mice and elephant flows. We found that network traffic can be clustered based on their performance metrics. As a result, each cluster contains similar traffic in terms of their packet loss, RTT, throughput, and so on. On the other hand, the challenges of identification of the traffic with big volumes (elephant) and short flows that transmit a small amount of traffic (mice), which comprise 90% of traffic on the Internet still exists. Therefore, identifying clustered traffic as elephants and mice can help to improve network management systems in the future and enhance the performance of a network. The work presented here can be applied by network operators to group the traffic based on their performance metrics and then identify them as mice and elephant, can help to find the optimal path for each flow, which in turn will contribute to improving network performance and QoS. The step of the identification of mice and elephant flows will be used as initial stages in our future work.

References

1. Suter, B., Lakshman, T.V., Stiliadis, D., Choudhury, A.K.: Design considerations for supporting TCP with per-flow queueing. Proc. - IEEE INFOCOM **1**, 299–306 (1998)
2. Mogul, J., et al.: DevoFlow: scaling flow management for high-performance networks. In: Conference Paper ACM SIGCOMM Computer Communication Review, pp. 254–265 (2011)
3. Gude, N., et al.: NOX: towards an operating system for networks. In: ACM SIGCOMM Computer Communication Review Submitt to CCR, vol. 38, no. 3 (2008)
4. Musumeci, F., et al.: An overview on application of machine learning techniques in optical network. IEEE Commun. Surv. Tutor. **21**(2), (2019). Second
5. Dong, S., Ding, W., Gong, J., Zhou, D.: Flow cluster algorithm based on improved K-means method. IETE J. Res. **59**(4), 326 (2013)
6. Singh, H.: Performance analysis of unsupervised machine learning techniques for network traffic classification. 2015 Fifth International Conference on Advance Computer Communication Technology, pp. 401–404 (2015)
7. Zhang, J., Xiang, Y., Wang, Y., Zhou, W., Xiang, Y., Guan, Y.: Network traffic classification using correlation information. IEEE Trans. Parallel Distrib. Syst. **24**(1), 104–117 (2013)
8. Zhang, J., Xiang, Y., Zhou, W., Wang, Y.: Unsupervised traffic classification using flow statistical properties and IP packet payload. J. Comput. Syst. Sci. **79**(5), 573–585 (2013)
9. Oudah, H., Ghita, B., Bakhti, T., Alrurban, A., Walker, D.J.: Using burstiness for network applications classification. J. Comput. Networks Commun. **2019** (2019)
10. Bujlow, T., Riaz, T., Pedersen, J.M.: A method for classification of network traffic based on C5.0 machine learning algorithm. In: 2012 International Conference on Computing, Networking and Communications ICNC 2012, pp. 237–241 (2012)
11. Lin, G.Z., Xin, Y., Niu, X.X., Jiang, H.B.: Network traffic classification based on semi-supervised clustering. J. China Univ. Posts Telecommun. **17**(SUPPL. 2), 84–88 (2010)

12. Terzi, D.S., Terzi, R., Sagiroglu, S.: Big data analytics for network anomaly detection from netflow data. In: 2nd International Conference on Computer Science Engineering UBMK 2017, no. January 2017, pp. 592–597 (2017)
13. Gratian, M., Bhansali, D., Cukier, M., Dykstra, J.: Identifying infected users via network traffic. *Comput. Secur.* **80**(October), 306–316 (2019)
14. Al-Saadi, M., Ghita, B.V., Shiaeles, S., Sarigiannidis, P.: A novel approach for performance-based clustering and management of network traffic flows. In: 2019 15th International Wireless Communication Mobile Computer Conference (IWCMC). IEEE Publication (2019)
15. Chhabra, A., Kiran, M.: Classifying Elephant and Mice Flows in High-Speed Scientific Networks, Prepr. Submitt. to INDIS, 19 September 2017
16. Ben Basat, R., Einziger, G., Friedman, R., Kassner, Y.: Optimal elephant flow detection. *IEEE Trans. Cloud Comput.* (2019). <https://doi.org/10.1109/TCC.2019.2901669>
17. Wu, X., Yang, X.: DARD: distributed adaptive routing for datacenter networks. In: Proceedings - International Conference on Distributed Computing Systems, pp. 32–41 (2012)
18. Wang, W., Sun, Y., Zheng, K., Kaafar, M.A., Li, D., Li, Z.: Concise Paper: Freeway: Adaptively Isolating the Elephant and Mice Flows on Different Transmission Paths (2014)
19. Cui, W., Yu, Y., Qian, C.: DiFS: distributed flow Scheduling for adaptive switching in FatTree data center networks. *Comput. Networks* **105**, 166–179 (2016)
20. Ostermann, S.: tcptrace. <http://www.tcptrace.org/>. Accessed 01 Oct 2018
21. “A Story of Mice and Elephants: Dynamic Packet Prioritization|No Jitter.” <https://www.nojitter.com/story-mice-and-elephants-dynamic-packet-prioritization>. Accessed 27 Aug 2020



Automated Generation of Zigzag Carbon Nanotube Models Containing Haeckelite Defects

M. Leonor Contreras¹(✉), Ignacio Villarroel², and Roberto Rozas¹

¹ Faculty of Chemistry and Biology, Environmental Science Department, University of Santiago de Chile, Usach, Avda. L. B. O'Higgins 3363, Santiago, Chile
leonor.contreras@usach.cl

² Faculty of Engineering, Computing and Informatics Department, University of Santiago de Chile, Usach, Avda. Ecuador 3659, Santiago, Chile

Abstract. Carbon nanotubes, CNTs, are a valuable material with applications in areas such as electronics, mechanics, optics and biomedicine, where they have great potential in the diagnosis and treatment of cancer, due to their ease of functionalization and their size that allows them traverse biological membranes and anchor on cancerous tumors. Regular CNTs have six-membered rings. According to their ordering or chirality, *armchair*, *chiral* and *zigzag* CNTs are known, with different properties. There are also CNTs that contain four-, five-, seven- and eight-membered rings called defects such as bumpy, haeckelite (Hk), and Stone-Wales (SW) defects. Defects modify the properties of CNTs. For example, *zigzag* CNTs with bumpy defects exhibit stronger interactions with the anticancer drug doxorubicin, DOX, than regular CNTs, facilitating drug transport to tumors. Designing defective CNTs as drug delivery systems would benefit from calculating DOX-CNT binding energies for CNTs of different chirality and with different defects placed at different positions, in an artificial intelligence approach. Such a systematic study requires the automated generation of defective CNTs. That is the goal of this work focusing now on *zigzag* CNTs with Hk defects. It starts from the step-by-step generation of the regular nanotube, leaving the positions where the defect will be inserted marked. Then a pair of carbon atoms is inserted and the corresponding bonds are completed, which ends with a quick optimization to visualize the nanotube structure. For this, a tcl script was developed with outputs compatible with many calculation methods and with possibilities of extension to other systems.

Keywords: Defects · Carbon nanotubes · Generation software

1 Introduction

Artificial Intelligence, AI, has become an essential part of various activities in current life, particularly when it is necessary to handle and process a large amount of data, due to its speed and effectiveness in obtaining representative trends of some properties, in different fields of action such as drug discovery [1, 2].

Carbon nanotubes, CNTs, on the other hand, have shown a great variety of stable structures with different chiralities, such as *armchair*, *chiral* and *zigzag*, conductive and semiconductive, of different diameters and lengths, single or multilayer. These CNTs, called regular, are made up of only six-membered rings. They have varied technological applications. Their molecular size allows them to cross biological membranes [3], with the advantage that they can be functionalized with groups that give them greater solubility and groups that can be specifically anchored in cancerous tumors [4]. These properties and their ability to adsorb different compounds make them important candidates to form drug delivery systems.

There are also CNTs that present rings of four, five, seven and eight members or combinations of these rings, called defects, which have different properties than the corresponding regular CNTs. Theoretical studies show some trends in the diameter of CNTs that favor interactions with doxorubicin, DOX, a recognized anticancer drug [5, 6]. Other studies indicate that the position of the defects affects the properties of the nanotubes [7] and that the number of defects also affects their properties [8]. These and other works indicate that DOX-CNT interactions can be controlled depending on several factors. Among the variables for analyzing the structure of nanotubes, and therefore their properties, are chirality, diameter, length, type of drug adsorption (surface or encapsulation), type of defect and both their number and position in the nanotube. A systematic study of these nanotube structure parameters and how they affect DOX-CNT interactions constitutes fundamental information in the design of drug delivery systems. For a first theoretical approach to calculating the properties of nanotubes with defects and intermolecular forces involved in its interaction with the drug, it is required to have the atomic coordinates of each atom of the system. This requirement should be performed in a quick and reproducible way through an automated generation of 3D models for a variety of possible defective nanotube structures. However, to the best of our knowledge there is no computer system with these capabilities.

Current computational tools allow the generation of various regular structures of CNTs and do not offer an alternative for the automated generation of defects [9, 10]. Defects have to be done manually, starting from the regular nanotube, which means a slow work, not exempt from errors and must be done by an expert person. We think that having an automated tool that allows the generation of various models of CNTs, even by non-experts, can be of great help in the design of new drug delivery systems, and this is what motivates the present work.

1.1 Purpose

Our goal is to contribute to the design of anticancer drug delivery systems based on carbon nanotubes using artificial intelligence techniques, AI, for which the first thing that is required is the automated generation of different nanotube model structures of different chirality with or without topological defects of different types that can be *i.e.*, single-walled CNTs containing “bumpy” [11, 12], haeckelite, Hk, [13] and Stone-Wales, SW, [14] defects. Figure 1 shows some CNT chirality examples and Fig. 2 depicts CNT defects.

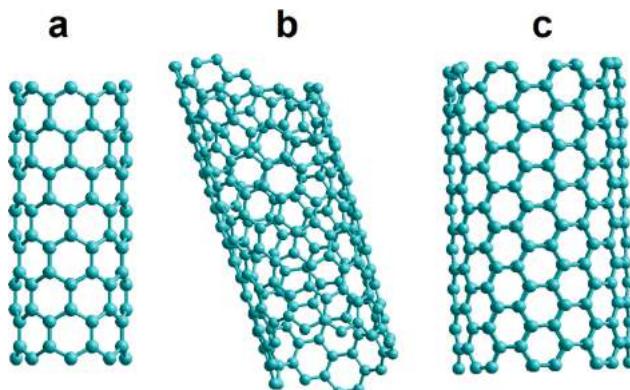


Fig. 1. Representation of CNT chirality examples. (a) *Zigzag* CNT; (b) *Chiral* CNT; (c) *Armchair* CNT.

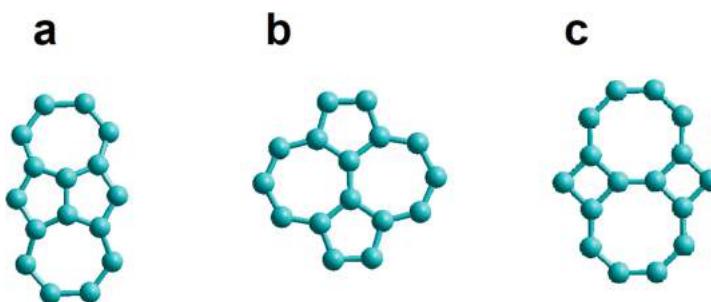


Fig. 2. Representation of CNT defects. (a) Bumpy defect; (b) Stone-Wales defect; (c) Haeckelite defect.

1.2 Background

Currently, although some computer systems have the capacity to generate carbon nanotubes of different chirality, these structures correspond to nanotubes that do not consider the presence of structural defects [9, 10, 15]. The computational study of these structures requires having the type of atom and its coordinates for each of the constituent atoms of the CNT with or without defects, in order to optimize its structure and calculate its electrical and structural properties. A systematic study requires the generation of many and varied CNT structures with and without defects, to evaluate their efficacy as carriers of antineoplastic drugs. That goal will be facilitated by using AI methods in their study, which evidently makes the automated generation of appropriate models necessary. The outputs of these systems should be in formats compatible with current program systems for calculating molecular properties and interactions. The first working approach and one of the simplest, consists of building a nanotube generation script that is compatible with a proven optimization and visualization system that has been shown to be efficient [16]. A similar approach has been used to generate *armchair* nanotubes containing bumpy and other similar defects [17].

This work specifically develops and implements a script that automatically generates models of open zigzag nanotubes with or without haeckelite defects. It also analyzes how this research tool facilitates the computational design of various models of nanotubes not yet known, delivering the Cartesian coordinates of each atom, in a format compatible with the calculation systems of computational chemistry, thus facilitating the study of their properties.

2 Method

We worked with a script designed in tcl language that integrates with the desktop molecular design program, HyperChem [16] in a modular way. That allows to add different independent functionalities, throughout the development, with the advantage that each function does not interfere with another and, in addition, the development is scalable over time.

The beta version of the generation of regular nanotubes has defined processes that consist of building simple hexagon units that are bound to form a ring up to the diameter defined by the user. Once the generation of the ring is finished, the procedure is repeated generating other rings that are linked to the previous one until reaching the length defined by the user, as depicted in Fig. 3 which also shows the stage of manual inclusion of a defect in the nanotube.

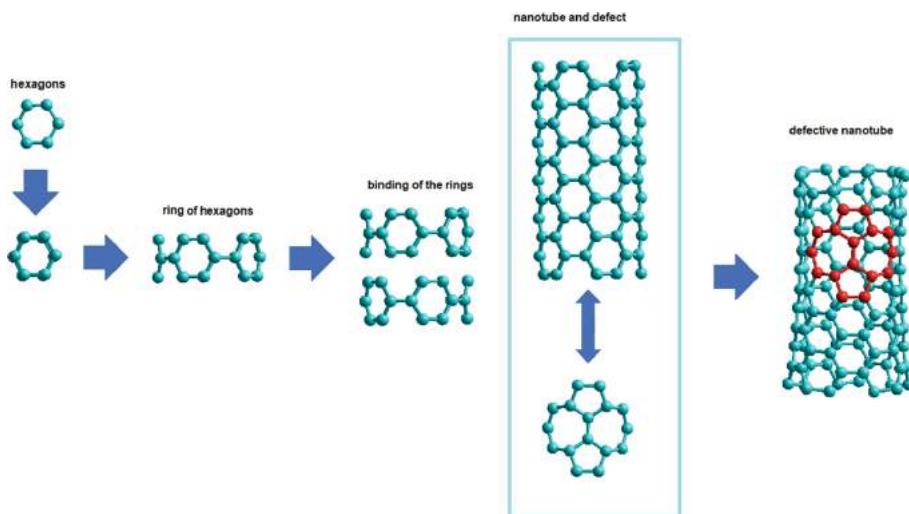


Fig. 3. General processes involved in the generation of defective nanotubes

On the other hand, for the automated generation of defects, in the first place, during the generation process of the regular nanotube just described, each atom that is created is defined through indexes. In the defect generation process, the indexes of the atoms in which the desired defect will be positioned are identified, leaving certain bonds vacant, so that at the end, when the nanotube generation ends, the respective bonds are completed

according to the defect. In other words, an integrated generation process is produced that allows the automated generation of the defective nanotube in a single process. Different is the case in which the procedure is done manually, since the regular nanotube must be generated first and then, at a later stage, the defect must be incorporated manually.

3 Results

An automated generation of *zigzag* CNTs of different dimensions that contain haecelite defects was developed by means of a tcl script. The script was installed in the Hyper-Chem molecular design program that allowed the optimization and visualization of the nanotube. Generation of Hk defects is produced by adding a pair of carbon atoms (ad-atoms) in the selected region during the initial stage. The ad-atoms are then bound to the nanotube atoms that were marked (and are kept without completing their bonds) in the initial generation process of the regular nanotube, as clearly illustrated in Fig. 4, which shows the different steps that are being accomplished. Step 1 shows the *zigzag* nanotube with the atoms marked and without completing their bonds, where the defect will be positioned. Step 2 shows the end of the generation stage of the regular nanotube and the generation of the pair of ad-atoms that will be attached to the nanotube. Step 3 shows the bonds formed. Step 4 shows the final Hk defect (4-8-8-4) where the four-membered and eight-membered rings are well visualized.

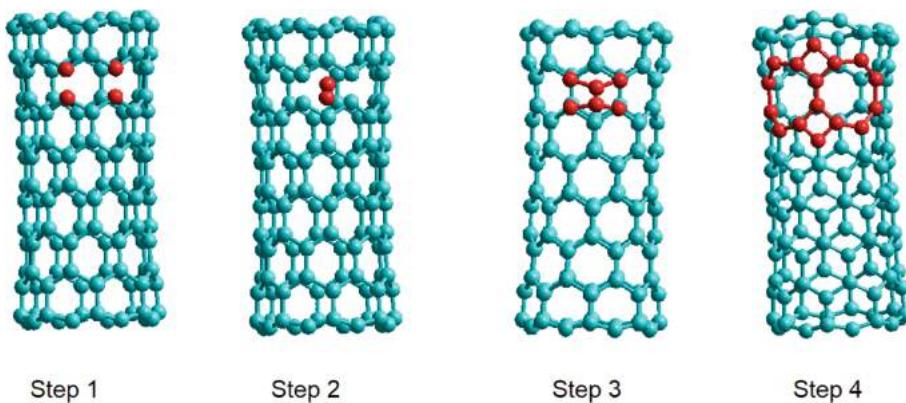


Fig. 4. Different Steps in the Automated Generation of Defective Zigzag Nanotubes. Step 1: the Bonds of the Selected Atoms are not Generated. Step 2: the Generation of the Nanotube is Finished and the Ad-Atoms are Generated. Step 3: Bonds are Built. Step 4: Optimization.

In computing there are two main paradigms which are structured programming and object-oriented programming that is the most recent. In this work it was decided to use structured programming, because despite being an older paradigm, it has substantial advantages in terms of processing speed, ease of understanding and reduction of code maintenance costs. The method of creating algorithms based on the selected paradigm is developed based on functions in order to reuse code and functions throughout development, which is also recommended in good programming practices.

The advantage of working with script at this stage is that you have the option to expand to new functionalities. It contains a set of algorithms that can be increased over time, externally, and in turn, can interact with functions of the HyperChem program.

One limitation is currently given by the language used (tcl) compatible with Hyperchem, since there is little documentation of it, with an undefined integrated development environment (IDE). Better languages exist in programming environments but they are not designed for defect-specific creation or optimization of nanotubes. Other useful systems in molecular design use Python scripts but do not generate nanotube defects in an automated way [15].

As future work, it is proposed to extend the automated generation of *armchair* and *zigzag* nanotubes with defects to *chiral* nanotubes. The generation of many different DOX-CNT structures will allow calculations of their binding energies, equilibrium distances and other properties which can be organized and analyzed with AI techniques in the proposal of new drug delivery systems.

4 Conclusions

We have developed an automated method of generating *zigzag* nanotubes of varying diameters and lengths containing haeckelite defects, as a fast and efficient strategy of having many structures of CNTs. The objective is to contribute systematically to the study of the properties of these new compounds as carriers in drug delivery systems.

Although this development is necessary to reach the molecular design of drug delivery systems through the methods of computational modeling and artificial intelligence, we think that it is both a great challenge and a good help that will reduce the costs and time required in the development of pharmaceutical molecular design and other related fields.

Acknowledgments. This work was partially supported by the Direction of Scientific and Technological Research DICYT-USACH Project Nr. 061941CF and by the Sociedad de Desarrollo Tecnológico SDT-USACH Project Nr. CIA 2981.

References

1. Zhu, H.: Big data and artificial intelligence modeling for drug discovery. *Ann. Rev. Pharmacol. Toxicol.* **60**, 573–589 (2020)
2. Hecht, D.: Applications of machine learning and computational intelligence to drug discovery and development. *Drug Dev. Res.* **72**, 53–65 (2011)
3. Liu, J.Z., Hopfinger, A.J.: Identification of possible sources of nanotoxicity from carbon nanotubes inserted into membrane bilayers using membrane interaction quantitative structure-activity relationship analysis. *Chem. Res. Toxicol.* **21**, 459–466 (2008)
4. Saliev, T.: The advances in biomedical applications of carbon nanotubes (2019). <https://www.mdpi.com/2311-5629/5/2/29/htm>. Accessed 14 Nov 2020
5. Wang, Y., Xu, Z.: Interaction mechanism of doxorubicin and SWCNT: protonation and diameter effects on drug loading and releasing. *RSC. Adv.* **6**, 314–322 (2016)

6. Contreras, M.L., Torres, C., Villarroel, I., Rozas, R.: Molecular dynamics assessment of doxorubicin–carbon nanotubes molecular interactions for the design of drug delivery systems. *Struct. Chem.* **30**(1), 369–384 (2019). <https://doi.org/10.1007/s11224-018-1210-5>
7. Rafiee, R., Mahdavi, M.: Molecular dynamics simulation of defected carbon nanotubes. *Proc. Inst. Mech. Eng. Part L: J. Mater.: Des. Appl.* **230**(2), 654–662 (2016)
8. Torres, C., Villarroel, I., Rozas, R., Contreras, M.L.: Carbon nanotubes having haeckelite defects as potential drug carriers. Molecular dynamics simulation. *Molecules, Spec. Number Comput. Methods Drug Discovery Des.* **24**(23), 4281–4305 (2019)
9. JCrystalSoft, Nanotube Modeler. 2005–2018 <http://www.jcrystal.com/products/wincnt/>. Accessed 14 Nov 2020. Melchor, S., Dobado, J.A.: An algorithm for connecting two arbitrary carbon nanotubes. *J. Chem. Inf. Comput. Sci.* **44**, 1639–1646 (2004)
10. Veiga, R.G.A., Tomanek, D., Frederick, N.: TubeASP. Carbon nanotube generation applet. <http://www.nanotube.msu.edu/tubeASP/>. Accessed 14 Nov 2020
11. Charlier, J.C., Ebbesen, T.W., Lambin, P.: Structural and electronic properties of pentagon-heptagon pair defects in carbon nanotubes. *Phys. Rev. B* **53**, 11108–11113 (1996)
12. Stenberg, M., et al.: Carbon ad dimer defects in carbon nanotubes. *Phys. Rev. Lett.* **96**, 75506 (2006)
13. Terrones, H., Terrones, M., Hernández, E., Grobert, N., Charlier, J.C., Ajayan, P.M.: New metallic allotropes of planar and tubular carbon. *Phys. Rev. Lett.* **84**(8), 1716–1719 (2000)
14. Stone, S.J., Wales, D.J.: Theoretical studies of icosahedral C₆₀ and some related species. *Chem. Phys. Lett.* **128**, 501–503 (1986)
15. Samson, Molecular design program. www.samson-connect.net/. Accessed 14 Nov 2020
16. HyperChem release 7.5 Hypercube Inc 1115 NW 4th Street Gainesville Florida 32601 USA
17. Contreras, M.L., Avila, D., Alvarez, J., Rozas, R.: Computational algorithms for a fast building of 3D carbon nanotube models having different defects. *J. Molecular Graph. Model.* **38**, 389–395 (2012)



Artificial Intelligence Against Climate Change

Leila Scola^(✉)

School of Engineering, Santa Clara University, Santa Clara, USA

Abstract. The industrial, transportation, and residential sectors draw the most energy in the United States. With most energy created by burning fossil fuels, a highly inefficient method of energy creation, global greenhouse gas levels are rising, raising the temperature of the earth, causing natural processes to become unbalanced. The health of the earth is declining. The rise of technology and persisting growth of computing devices known as the Internet of Things (IoT) and increasing automation of systems through Artificial Intelligence (AI) and Machine Learning (ML) is a factor of energy expenditure as more humans desire devices and more systems are built. The ethical implications of utilizing new technology should be evaluated before creating more. This paper explores modern computing systems in the sectors that draw the most energy, and, more specifically, the role AI and IoT play in them. Each sector may become more energy efficient, productive, and safer by introducing edge computing through IoT devices and coupling it with AI computing abilities that already automate most processes. Multiple studies show energy consumption and costs are lowered when edge computing is paired with the IoT and AI. There is less human involvement, more regularity in execution and performance, and more widespread use because of the accessibility. This creates safer, cheaper, energy-efficient systems that utilize existing technology. The ethical implications of these systems are much more positive than what already exists. Coupling the power of AI with the IoT will reduce energy expenditure in modern systems and create a more sustainable world.

Keywords: Artificial Intelligence · Internet of Things · Machine learning · Ethics · Sustainability · Energy efficiency · Edge computing · Fog computing

1 Introduction

In the United States, certain sectors – industrial, transportation, and residential – draw the most energy. These sectors are having actuators and sensors increasingly integrated into their increasingly automated management systems. Global warming has also been increasing due to a rapid burning of fossil fuels in the last century, releasing atmospheric CO₂. The increase in energy consumption was

closely linked with the inception of the Industrial Revolution, which did revolutionize the tools humans have access to. It is unrealistic to ask humans to give up the tools and technology we rely on for daily activities, so we should utilize the technology to stem the rise of climate change. By effectively using Artificial Intelligence (AI) and the Internet of Things (IoT) we can use energy as efficiently as possible, allowing us to keep our technology and create a sustainable future.

The detriments of climate change are perilous and imminent. The average global temperature has risen 1 °C since the Industrial Revolution and it has been estimated that if it increases by one more degree, we will reach a tipping point – a point of climate change of which there is no return or reversing of the damage on the earth. It will lead to higher temperatures across the globe. The equator will continue to warm, causing the land that is normally arable to be pushed northward, where unfortunately soil is either lacking nutrients or soil is rocky. The occurrence of natural disasters, such as forest fires and hurricanes, will increase. As temperatures rise, glaciers will melt, the rainy season will be shorter, and drinkable water will decrease. Without arable land and water, we will not be able to produce enough food to feed the inhabitants of the earth. This change is set to happen by 2035 if we continue at our current rate of energy consumption and fossil fuel burning. It is our responsibility as humans to stop the damage we have imposed on this shared habitat. We do not have a second earth.

Current solutions to mitigate climate change are either very niche or very small, therefore ineffective in altering global infrastructure in the ways we need to truly stop global warming. Individuals have the ability to make choices about recycling, or using reusable napkins and other small changes, but changes on an individual level will not make a huge difference. In addition, some government subsidies are offered for employing more sustainable practices, such as using certain fuel sources. Unfortunately the cost of continuing practices that increase global warming are usually cheaper than finding an alternative. There have been individual efforts by large tech companies to employ practices that utilize technology, but these are fairly niche. Google DeepMind has been used to reduce the energy that cools its data centers and Microsoft SilviaTerra which tracks the health of forests [16]. While these solutions are effective, they do not influence the infrastructure of modern economies and households in ways that will stop global warming. In addition, developed countries have the luxury of affording alternative solutions, whereas developing nations are relying on cheap methods of achieving success, which usually incorporate the use of fossil fuels in ways that are rapidly increasing greenhouse gas emissions. By changing the way devices in the IoT are used by their AI counterparts, we can positively alter all modern systems and infrastructures to reduce their energy consumption.

In our modern era of technology, there are sensors everywhere tracking the status of our environment and helping us make decisions. This abundance of computing devices with the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction is known as the Internet of Things. Artificial Intelligence replicates human intelligence by interpreting

data to achieve a human-programmable goal. Coupled with modern hardware technology, it can process thousands of data points rapidly.

AI technology is currently used to automate systems around the world, such as thermostats in homes or assembly lines in factories, by processing mass amounts of data collected from sensors from a selected area using human-programmed algorithms, detecting patterns that may go unnoticed by humans. The IoT is rapidly growing as we create more sensors, automators, and devices that can track more system parameters than ever before. The trends in automation and device creation will continue. Technology pervades everything in the modern world. Rather than oppose technological development that has driven human innovation for the last few decades to reduce energy expended creating new products, we should capitalize on existing technology to reduce energy waste and improve efficiency of modern systems. In this paper, we will explore how AI's processing capabilities coupled with the growing IoT can be utilized to identify areas and patterns of wasted energy and autonomously correct wasteful practices while adhering to human standards of comfort, in turn mitigating climate change to an extent humans are not capable of doing on their own.

Despite IoT, AI, cloud computing and edge computing all generating CO₂, we can leverage these technologies to not only reduce their own energy consumption, but the systems they are embedded in as a whole. It is unrealistic to reverse our reliance on these technologies, so we should instead use them effectively. In the rest of this paper we will outline energy consumption the United States, systems that control the industrial, transportation, and residential sectors, and how they may be improved upon to stem our global greenhouse gas production, as well as the ethical implications of using technology in this way.

2 Background

In this section, we describe where the most energy is consumed in the United States and how AI and the IoT function.

2.1 Energy Consumption in the United States

The residential, transportation, and industrial sectors draw the most energy in the United States [11]. Their distributions can be observed in Table 1. Additionally, two-thirds “of anthropogenic greenhouse gases (GHGs) emission originates from fossil fuel combustion in the transportation and industrial sector” [7]. The transportation sector alone has become the second largest contributor of CO₂ emissions, which have currently posed the most serious problem to the environment as the biggest contributor to climate change [7]. Meanwhile, industries consume the most power in any country and “buildings in use or under construction are the greatest single indirect source of carbon emissions accounting for 50% of total emissions” [8, 10].

Currently, electricity is used to power industrial settings, transportation networks, and residential communities (as well as commercial enterprises). Due

Table 1. Energy drawn in the united states by sector[11]

Sector	AI managing system	Energy drawn in U.S.
Industrial	32%	Industrial IoT (IIoT)
Transportation	29%	Energy Management System (EMS)
Residential	20%	Building Management System (BMS)
Commercial	18%	Management System (MS)
Other	1%	—

to current methods of production and the makeup of the fuel mix, electricity is highly energy inefficient. It is delivered at about 30% efficiency since the primary energy contained in the fuel is only partially exploited [10]. Automated systems that work with the IoT to manage energy distribution and use in buildings, transportation networks, and industrial settings will reduce energy consumption.

2.2 AI and the IoT

The sectors mentioned require dynamic power management and contain ample sensors and actuators – IoT – that can be utilized by AI [8]. AI can collect the vast range of data from these settings, can determine energy patterns, and can then implement human-specified goals based on energy use trends.

AI that is hardware-accelerated is capable of efficiently processing a multitude of data while the IoT is capable of providing an abundance of data. Artificial Intelligence is generally understood to be able to mimic aspects of human intelligence in its computations. This includes planning, learning, problem solving, decision making, pattern recognition and more. Generally, through the use of training data and specific algorithms, AI is able to be applied to varying scenarios to carry out useful computations. Together, AI and the IoT, may identify patterns humans cannot discover easily or possibly at all often due to the large quantity of data to sort through or the complexity of the patterns created by interacting factors in a system. Machine Learning (ML) is a subset of AI that is able to learn relationships from large datasets it is trained on [2]. It enhances AI in that it allows AI to “learn” without specific programming or algorithms. Essentially, ML allows a system to gauge the accuracy of its results to recycle the useful data, parameters, and computations back into its own algorithm to refine its computations to become more accurate in its intended purpose. ML can find linkages between locations, times, and quantities in datasets and AI can build on ML connections to provide automated warning and advice [2]. AI and ML use different modelings based on the human-programmable goal to be reached, as can be observed in Table 2.

Table 2. Types of mathematical modeling and their correlating use

Types of mathematical modeling	Purpose	Example of ML applications
Differential equations	Continuous and coupled to account for parameters in a system that may be intertwined	Climate system [2]
Nonlinear equations/ Regressions	Allows encapsulation and tracking of many parameters	Climate system [2,4]
Nonlinear, Non-Gaussian inferences	Models randomly changing systems based only on current state	Climate system [2]
Kalman filtering-based gray box	Predict and determine statistical process control limits for fault detection	HVAC systems [3]
Q-learning	Identifies best action given current state	Lighting Control System [3]
Directed graphs	Allows parameters to be given varying importance to uncover relationships in nonlinear data	ESM [2]
Gaussian models	Models systems with many parameters that may be unknown	Tracking headcount in rooms [3]
Linear regressions	Used for resource prediction, response time, throughput and CPU utilization	Choose best edge node [9]

2.3 Importance of Mathematical Modeling

As mentioned, Machine Learning enhances Artificial Intelligence by providing a way to mutate algorithms to become more responsive to their systems and more accurate in their computations. There are several ways to achieve this depending on what the system is intended to be used for. Depending on system use, there are best practices for identifying relevant data, collecting relevant data, feeding the data back into the system, and revising the algorithm. In this section, we will describe how this is achieved in some detail.

Because algorithms are originally written by humans in some capacity, they are therefore based on mathematical theories and principles. We devise mathematical models to model system behavior and data flow. ML algorithms take initial mathematical models and revise them for better functioning within the system. In Table 2, several Mathematical Models and their use are explained.

To begin, differential equations are often utilized in making predictions about how climate systems will behave, allowing us to predict what response our system should make to preserve a comfortable space for humans while remaining energy efficient. This is because differential equations allows for coupling any source and sink terms, remaining robust through dimension reductions and changes in

the system, giving insight to interconnections within a complex system – the climate network – that humans would not be able to track or model themselves. As best stated by Huntingford, “knowledge of controlling processes points to model parameters that most strongly influence projections, guiding measurement campaigns to aid uncertainty reduction” [2].

Nonlinear equations and regressions serve a similar purpose in terms of understanding the system, similarly allowing for the encapsulation and tracking of many parameters in a complex system – something linear models would be unable to do [2,4]. This aids us in understand complex and variable climate systems.

Non-linear, Non-Gaussian Models are useful for ML prediction and inferences when we are dealing with spatio-temporal processes, since climate systems contain many state-space problems where mechanisms are latent, or hidden, so only resulting data is observable and available.

The Kalman Filtering system is a subset of Non-linear, Non-Gaussian Models that is slightly simpler than other models, which can help in subset systems within the climate network.

Q-learning differs from previous mentioned mathematical models in that it is a model-free learning algorithm. It is useful in systems that are non deterministic and irregular, telling agents what actions to take in certain situations without an environment model. For that reason, it can be used to control lighting control systems, often turned on and off randomly. It can be used to manage light intensity, identify daylight trends, and used in conjunction with occupancy-detection [3].

Directed graphs are often used in conjunction with other methods, but are useful alone as well. Directed graphs model system parameters as nodes with connecting edges that have associated weights, or ‘biases’. The larger the bias, the stronger the connection, showing how certain parameters interact with one another, showing trends within the ESM. These are often used with neural networks, where each node makes up a neuron. Using deep learning or neural net approaches avoid specifying a process and instead work with data to improve understanding or multivariate relationships in nonlinear systems. Upon each iteration of data through the system, weights are updated to better reflect system interaction [2].

Gaussian models are similarly to neural networks in that it works on data and observable functions rather than input state. A Gaussian process is a collection of random variables such that a subset of them has a multivariate normal distribution. The process “is specified by a mean function and a covariance matrix” [2]. Since it is a non-parametric approach, it allows for direct representation of uncertainty and prior beliefs, powerful in nonlinear regression analyses. Predicting capability past sample inputs is necessary for predicting future states of systems.

Lastly, linear regressions are some of the most simple mappings between variables. It models the relationship between a response and at least one related variable. It is often used to track resource prediction, response time, throughput,

and CPU utilization. These are easy to track since they are physical node performance variables based on data flow. Tracking node with best performance allows us to choose the best node for computation purposes in Industrial Systems.

We have not included many mathematical models in this short discussion, however, we aim to emphasize the diversity of models available and their varied purposes and applications to the sectors that could utilize AI and ML to increase energy efficiency. Mathematical models help process data most efficiently in any system. The existence of many show that despite the variation in systems and circumstances, there is an existing model that can be utilized to perform energy efficiency computations effectively.

2.4 Summary

Different systems warrant different tracking mechanisms. There are multiple models and equations that may be used for a system, varying in use based on the parameters to be tracked and outcome AI is expected to produce, often selected by humans running the system. Humans are beginning to use AI and the IoT in effective ways to achieve goals with more speed and precision than ever before.

3 Existing Technology

3.1 Buildings

BMSs consist of low-cost personal and hyper-local sensors that track user habits and utilize green energy and storage resources for environmental sustainability while implementing smart systems with autonomous decision making [3]. They are cost-effective, energy-efficient and maintain user comfort. This is becoming increasingly important as demand for energy increases while dominant energy supplies continue to dwindle.

The IoT allows cost-effective integration of sensors to monitor and identify different energy and environment-related parameters to determine building's health, energy requirements, and electricity usage behavior of subsystems [1,3]. Energy consumption data can be obtained from modules monitoring and controlling a variety of physical conditions [1]. Data can consist of humidity, temperature, user preference, weather predictions, human activity in certain locations, etc. The data and correlating parameters are transmitted over a wireless local area network (WLAN) or a wireless personal area network (WPAN), often to a central control station where the energy management logic can instruct specific calculations to provide the users with an energy management solution.

AI suggests changes that can be implemented autonomously if given user permission to correct energy-inefficient trends [1]. Current Building Management Systems (BMSs) logic can be seen in Fig. 1. For example, if the system notices fans are left on when humans are not in the room, it may turn them off. Current technology ranges from AI-based Energy Auditing, Monitoring and Control for

Utilities, to Multi-model Climate System Networks, to complete BMSs. Despite varying levels of hierarchy, all residential systems need large volumes of data to identify user trends to function effectively.

Processor availability, speed, connectivity and cheap data storage has increased ML and AI use, and improved hardware has helped with memory and computation, increasing speed and efficiency more than ever before [2]. Because climate change analytics are data-intensive, ML algorithms could not be implemented until recently due to computation architecture and power restraints. However given recent technological developments, BMSs provide an inexpensive, user-friendly system capable of easy installation that will conserve energy in a variety of settings while utilities are used to monitor the efficacy of energy efficiency [1].

3.2 Transportation

Transportation systems are littered with sensors, actuators, and communication devices to allow for mitigating problems, traffic, congestion, and CO₂ [7]. Every transportation network is filled with sensors, lights, cameras, speed monitors, etc. each able to monitor and report data. AI, blockchain, and cloud computing has made transportation networks grow rapidly. The launch of several IoT applications and frameworks has improved transportation methods while reducing short and long term negative impacts. These improvements in technology include navigation applications, fuel monitoring software, more efficient intersections, calculation of better city planning, etc. By planning for the future, less time is wasted in traffic, on inefficient routes, or by inefficient vehicles.

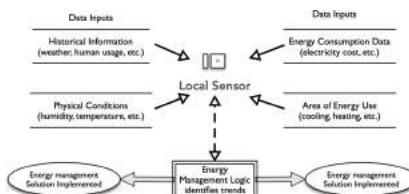


Fig. 1. How current BMS technology executes

By integrating data collected from the IoT into Energy Management Systems (EMSs), vehicles can optimally utilize energy while interacting with driver behavior and accounting for the condition of environmental impacts, also making data visible to users [6, 7]. How transportation networks function can be observed in Fig. 2. Increasing the efficiency of real-time data flow and information quality allows us to establish intelligent transportation systems and traffic networks that increase safety and energy efficiency [7].

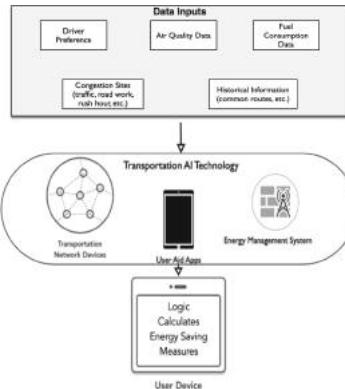


Fig. 2. How transportation networks function

3.3 Industry

The industrial sector is beginning to utilize the IoT and AI to full potential, known as the industrial IoT (IIoT) – see Fig. 3 [8,9]. This sector requires energy efficiency and real-time data to create safe, productive, reliable systems. Industrial systems are often very large and therefore have many data points to collect and, often, machines to run. This influx of data is most commonly sent to the cloud, where calculations are run and directions to continue system performance are sent back to actuators. The magnitude of data sent to the cloud is creating unacceptable forms of latency (time taken for information to travel from one node to the next), packet drops (information that does not send), and failed throughput (amount of packets sent in a given time period), decreasing scalability, reliability, and functionality.

AI and ML can make industrial systems more intelligent, dynamic, flexible, and scalable by creating a networked and remotely accessible system of the industrial machinery [8]. Energy consumption of sensors and adaptations in frequency of data generation can be monitored and implemented with improving Wireless Sensor Networks (WSNs) to facilitate fault and resource prediction, quality management and product development and arrangement [9]. AI can effectively process and control the magnitude of data generated by the growing IIoT.

An emerging form of computing known as edge computing consists of data being sent to nodes close to sensors, rather than the cloud, to reduce traffic, packet drops, and latency. In some of the most successful examples of edge computing, AI algorithms choose the best edge node by predicting cloud processing resource usage, predicting requests, and identifying the best processing module for incoming requests [8]. Optimization algorithms often aid this process. Algorithms also help with time synchronization while linear IoT models enhance prediction accuracy with iterative processing to reduce energy consumption compared to cloud computing, making the IIoT more effective and scalable [9].

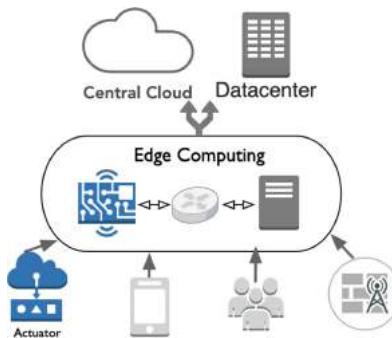


Fig. 3. How the IIoT functions

The linear IoT model saves energy through edge node reconfiguration and specific edge node selection [9]. Edge node reconfiguration allows nodes to reconfigure in certain time intervals to be able to process the varying types of services that occur in an industrial environment, making them more likely to process incoming requests [9]. Next, edge node selection conducted using the Shortest Estimated Latency First (SELF) algorithm reduces packet drops and latency in non-critical event systems [9]. Efficient edge nodes reconfiguration and selection for IoT nodes reduces energy consumption, uneven packet loss, and increases hit ratios at edge nodes.

4 Blockades in Technological Progression

The two main challenges slowing technological progression are a lack of cohesion within the engineering community in terms of sharing information and a lack of policy dictating what technology may or may not be implemented and where, outlined in Table 3.

Table 3. Outline of obstacles in AI and IoT development

Issue	Solution	Why we need it	Implemented by	What it provides
Lack of cohesion	Standard benchmarks for algorithms	Evaluating and comparing algorithms	OPEB framework	Thorough collaboration on AI algorithms to fix global issues
Lack of policy	Tool for comparing scenarios	Sustainable development	Game theory constraint-based model	Comparision of alternate scenarios

4.1 Lack of Cohesion

Standard benchmarks are needed to evaluate and compare algorithms. By comparing the efficacy and efficiency of algorithms, better decisions can be made about how to implement the algorithms, and more collaboration can occur to improve them. In particular, standard benchmarks will improve and lower the cost of AI use in continuous control tasks, such as self-driving control, urban transportation, and industrial robots [4].

AI-based control creates uniform processes, reducing minor variations in workflow and wasted energy, optimizing the process based on collected data points that can be fed back into the algorithm as the process executes to further reduce energy waste in a process known as reinforcement learning. However, without many AI benchmarks available and standard industry models, it is difficult to standardized typical agent actions and record best methods. In addition, without public standard benchmarks, AI testing environments are often kept secret or vary widely between testing platforms.

Given the cost and difficulty of AI research and the difficulty of comparing independent projects, further complicated by the fact that AI evaluates dynamic systems, progress is slow, uncollaborative, and may take varying paths of development by different groups [4]. As a result there is a lot of unpredictability of typical agent actions and reaching a useful model is often time-consuming and costly [4]. Additionally, there is a noticeable lack of formal proofs for closed-loop systems [4]. In this state of unpredictability, AI cannot function optimally.

Only wealthy companies can afford to develop AI algorithms in these conditions. A suggested solution is an Open Physical Environment Benchmark (OPEB) framework to share different physical environments, either through computer models or 3D printing designs, and lists of cheap materials to replicate testing scenarios. The development of a unified interface to share ideas and progress would allow for collaboration on and integration of different designs that address similar problems regardless of hardware, software, and implementation design. This collaboration would accelerate AI development and address challenges seen in real-world physical systems [4].

4.2 Lack of Policy

Policy-making is a complex process that occurs in adjusting environments and affects the three main pillars of sustainable development: economy, society, and the environment [6]. Currently, policies are created by setting objectives, which are constrained by budget and geophysical restraints. If problems are noticed after implementation of policies, only corrective measures can be applied. This is highly ineffective and less than optimal.

AI can aid with agent-based simulation, opinion mining, visual scenario evaluation, and optimization to support specific cases. By using equations from Game Theory Constraint-Based Models that define variables (activities), constraints (relating variables), and objectives, we can reach strategic equilibrium based on predicted user behavior [6]. Creating well-rounded policies faster by

comparing scenarios and weighing pros and cons would lead to the most sustainable development. The sustainable and beneficial policies that AI creates would not only effectively guide and streamline AI technology development and use, but also benefit our economy, society, and environment, making technology development as a whole more efficient and purposeful.

5 Developing Innovations

The potential of the IoT has been realized; therefore, technology is actively being developed that will use data collected from the IoT to improve AI and ML performance. Below are emerging designs in each sector that will improve systems and reduce energy consumption.

5.1 Buildings

Currently AI allows us to control specific devices within buildings, such as temperature, lights, humidity, shading, etc. This use of existing technology can be improved by connecting more devices from the IoT into BMSs, Multimodal Climate Sensor Networks (MCSNs), – similar to a BMS but often regulating fewer building systems – or even displayed on web interfaces to create more dynamic, efficient, useful systems.

As buildings get used, thermal characteristics deteriorate, usage patterns change, microclimates change, etc. and eventually performance of the building's system falls short of expectations. The IoT can allow cost-effective integration of sensors to monitor and identify different energy and environment-related parameters, as well as determine building health, energy requirements, electricity usage behavior of subsystems, and more [3]. Sensors can monitor ambient air pollutants, environmental parameters, historic trends, human preferences, cost of electricity, etc. Multiple parameters can be tracked at one time and relayed to a central processor. This information can also be broadcast to a platform that displays trends in a form akin to online social networks [5].

MCSNs can provide a networked way to monitor air quality and control building parameters. Similarly, BMSs can be thought of as cloud-based ecosystems that use social interaction to develop global patterns to achieve goals, use green energy and storage resources for environmental sustainability, and affirm establishment of smart systems with autonomous decision making [3]. Reference Fig. 4 for designs.

MCSNs and BMSs need a large influx of data to work effectively. They take data from sensors, relay it over a wireless network to a central processor, and determine energy-saving methods based on collected parameters. The decision is autonomously implemented into building systems. These decisions may range from turning off unused lights, actively regulating temperature as more bodies enter/exit rooms, storing energy unused energy, etc. By utilizing data that is already being constantly monitored and processed from IoT devices, building systems will become smarter, work more efficiently, and save energy and money.

For example, one study concluded that implementing a MCSN that regulates the HVAC machinery by set-point temperature based on human activity and room occupancy reduced the building's energy consumption by 33% [3].

As mentioned, information about parameter trends and energy usage can be put on a web interface. Here, communities may evaluate historic trends and recognize patterns between activity and pollution. Driven by social interaction, users increase awareness of air quality, share knowledge about the observation and management of personal air pollution, and build clean air reputations [5].

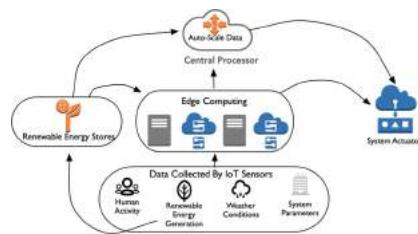


Fig. 4. Architecture for developing BMS that may potentially achieve net-zero-energy expenditure

Systems can be further improved by storing energy to be used later, pushing energy onto community grids, or drawing energy from the grid at different rates depending on the time of day to become even more efficient and less wasteful. These computations will not only be sped up, but will take less energy to execute if offloaded to edge nodes, rather than processed in a large central processor or cloud. Computing at a local level reduces congestion, latency, and packet dropping. Energy is saved by avoiding re-sending packets, having to execute large computations in one location, and having to expend energy when sending packets over multiple nodes. In turn, implementations of solutions will be more effective, quicker, and energy-efficient.

In conclusion, a zero-energy building may be achieved through high-performance design, integrated physical systems, a symbiotic building within its context, an interactive power grid with the building's energy generation system, and web interfaces [10].

5.2 Transportation

In the transportation industry, sensors are already integrated in vehicles and traffic networks, relaying data to centers that use AI to calculate and autonomously implement changes, saving energy. Every modern vehicle has a GPS system, speed sensors, and often energy efficiency gauges. Many intersections have cameras, traffic lights, and weight sensors. Depending on the type of networks, accompanying sensors vary. These sensors are a part of the IoT, all collecting information.

Dynamic EMSs could be developed to reduce traffic, pollutant emissions, and cost of transportation, while increasing transportation safety, comfort, and energy efficiency if this data was processed in a larger, interconnected network. User aid devices can be developed that both educate users and regulate transportation. AI can improve fuel efficiency by calculating best performance practices based on processed parameters of speed, energy consumption, and congestion. Efficient routes can be calculated for users. By increasing the monitoring of transportation networks, safety is improved, user comfort is maintained, and energy efficient trends directed by AI and IoT calculations.

Due to the rapid growth of the IoT, AI, blockchain, sensor technologies, cloud computing, and other technologies, data may be offloaded from vehicles into edge nodes for efficient and easy computation. Most transportation data is already offloaded to a remote location to store as history to save user preference, collect trends, and monitor safety. In addition, edge nodes could compute ways to improve the efficiency of transportation networks. Little, if any, additional energy would need to be used to modify existing systems.

Such an existing system of IoT and AI integration is the Intelligent Transportation System (ITS), the subset of IoT currently dealing with transportation. Road-side units can be equipped with edge technology to enable ITS. ITS could support in-vehicle entertainment, context-aware and location-aware services, smart parking, and smart traffic lights [8]. Despite challenges posed, coupling edge computing with the IoT and AI is expected to reduce CO₂ emission on a global scale by developing smart cities, dynamic transportation systems and electrical grids, implementing energy-saving gains [7].

Edge computing, combined with the IoT and AI, can address modern transportation issues due to the increasing number of sensors, actuators and communication devices that can reduce CO₂ emissions on a global scale by developing smart cities, transportation while mitigating traffic issues [7].

5.3 Industry

As mentioned, the rapidly growing amount of industrial data is beginning to be processed on edge nodes to allow for monitoring, processing, and controlling of critical and event triggering devices with ultra-low latency and necessary data storage. Wireless Sensor Networking (WSN) technology has evolved in such a way that it allows for “better quality management, energy efficiency, fault prediction, product planning, and resource prediction” by connecting sensors in industrial settings [8]. This creates cyber physical systems (CPS), extreme automation, smart factories, industrial robots, actuators and more. This “minimizes human error, lowers risk to human health, improves operational efficiency, reduces costs, improves productivity, and allows higher quality maintenance/customer satisfaction” [8].

In industry, edge computing is responsible for

- real-time industrial big data mining for high performance;
- concurrent data collection from multiple types of sensors, robots, and machines;

- fast processing of the sensed data to generate instructions for the actuators and robots within some acceptable latency;
- interfacing incompatible sensors and machines through necessary protocol translation and mapping; and
- managing system power management.

AI and ML have the power to make manufacturing more intelligent, dynamic, flexible and scalable by creating a networked and remotely accessible system of the industrial machinery – see Fig. 5 [8]. Because many physical sensors are costly and cannot keep abreast with the changing requirements of a factory, incorporating WSN and virtual sensing allows for flexibility and customization while keeping costs low. These networks monitor energy consumption of sensors and adapt the frequency of data generation, while also exploring and managing other energy sources (solar, thermal, etc.) [8].

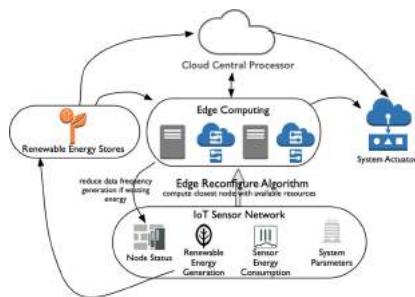


Fig. 5. Model for energy-saving IIoT system

By sending data that controls industrial processes and machinery to edge nodes to compute energy-saving strategies and most efficient practices, there has been a proven reduction in energy consumption, less use of machinery, and fewer human errors produced due to lack of human involvement [9]. Algorithms exist to help with time synchronization of machines and node computation to reduce energy consumption further, making the IIoT more effective and scalable. Reducing the use of huge machinery commonly found in industrial systems saves a lot of energy [8]. There is an overall increase in economic efficiency and, if other energy sources are managed as well, an increase in renewable energy use [8].

As with any system, new technology brings a lot of challenges, which can be converted to opportunities if better planning and standardization are done [8]. Systems that choose the best edge node exist, working in tandem with models that predict cloud processing resource usage, an AI algorithm to predict requests, and another AI algorithm to allocate incoming requests to the best processing module [9]. A linear IoT model enhances prediction accuracy with iterative processing. An ML algorithm calculates resource prediction in the cloud using linear regression and neural networking based on gathered data from the database of

the industrial process, the algorithm products response time, and CPU utilization. These algorithms predict the most likely incoming request, ready near-by nodes, process incoming requests, execute commands, and offload work to the cloud if the edge node cannot handle the size of the computation.

A study that monitored energy consumption and system efficiency showed a reduction in energy consumption and an increase in system efficiency when using the combined edge and cloud computing system, compared to the cloud computing model, which in itself reduces energy consumption compared to regular system processes. Combining AI, the IoT, and edge computing saves energy, reduces costs, and improves existing systems.

6 Ethical Implications of Emerging Technology

New technology brings an amalgam of new challenges, but these may be harnessed to create opportunities with proper planning. Technological advancement marks human life, so it would be unreasonable to expect humans to abandon the technology that makes our lives easier. Rather, we should harness it to create a more sustainable future.

In order to review the ethical implications of harnessing AI and the IoT in modern systems, we will first define some prominent ethical schools of thought, then consider how modern systems affect our planet and communities, then evaluate them from an ethical point of view. Prominent ethical theories include virtue ethics, consequentialist ethics, and deontological ethics. Virtue ethics asserts that humans should act in ways that cultivate good virtues [17]. Consequentialist ethics asserts the right path of action is the one that produces the most “good”, which can be evaluated in terms of pleasure, satisfaction, welfare, etc. [18]. Lastly, deontological ethics states that one must do what is “right”, which may not always produce the most “good” [19]. In short, the proposed system must maintain human autonomy to make decisions and produce positive effects [15].

Multiple case studies have shown that by utilizing AI and IoT technology, energy consumption of a system is reduced. The evidence is in BMSs, MCSNs, transportation networks, and the IIoT. While many of these are in early stages of development, a poignant example of the effect of AI and the IoT is by Google’s DeepMind’s ML which has been used to manage the energy consumption of Google Data Centers [11]. Despite energy usage growing 90% from 2000–2005, 24% from 2005–2010, and still slowly increasing, energy usage within the center has been kept flat and energy used to cool data centers has been reduced 40% [12,13]. This system implemented was able to keep track of the parameters of this system that behave nonlinearly. Traditional formula-based engineering and human intuition could not capture it, nor adapt quickly enough [12]. AI could operate without rules or heuristics of every operating scenario.

Energy consumption is rising and so is global warming. Global warming is ruining viable farmland, melting glaciers and the polar ice caps, and causing more natural disasters than ever before to name a few catastrophic issues. These issues

are directly correlated to the rising rate humans are using natural resources. Systems created by joining AI and the IoT not only mitigate, but reduce energy consumption, and therefore resource usage. As explained in sections above, we have already implemented the technology necessary to reduce energy consumption globally into modern infrastructure. The only change we need to make is ensuring these systems auto-regulate energy consumption while performing their jobs and maintaining human comfort. Algorithms to do so already exist. Implementing this solution takes very little human effort and produces the benefit of slowing, if not stopping, wasting of energy.

Evaluated in terms of normative ethical theories, this suggested solution is ethical. In terms of virtue ethics, utilizing IoT and AI in this manner will allow humans to create altruistic systems, therefore acting altruistically themselves. It also allows humans to develop themselves intellectually and socially as they collaborate with others to build these complex systems. By creating systems that reduce the production of greenhouse gases worldwide, we create a sustainable future. In terms of consequentialist ethics, this maximizing the overall “good” in the world, whether that is pleasure, satisfaction, welfare, etc. Longer, healthier, worry-free lives are positive on all accounts. Lastly, for the same reason this proposed solution is virtually and consequentially ethical, it satisfies the requirements of being deontologically ethical. Modifying modern systems to ensure we do not ruin our planet is the “right” thing to do, even if the change may initially be difficult. Combining the IoT and AI to mitigate global warming is ethical.

It allows humans to achieve more technological progression, reduce global warming, and create a sustainable future. This technology gives humans more options to choose the lifestyle and sustainability options that function best with their respective lifestyle. It also undoubtedly improves energy efficiency. Lastly, this technology allows humans to achieve more autonomy as more devote less time to managing systems, while, simultaneously, more jobs are created through rising productivity and developing systems [14]. It is an ethical improvement from any point of view. Harnessing this emerging technology will provide a better future for humans.

While some concerns may be brought up regarding security, privacy, and work created, this solution does not present any new concerns. In any computing system that processes data, there is a concern that data may be shared by malicious attackers or systems may be hacked to malfunction in ways that may harm society. The systems we are proposing are to process data to make them more energy efficient. If a hacker was to access data, they could do that as the system stands. If a hacker was to make the energy waste energy or resources, they could also make that change now. Wasting resources or causing a system to malfunction is extremely easy and can happen in a variety of ways. Implementing a solution that reduces energy consumption or produces energy reserves can only happen one way. If a malicious attacker was to stop these positive solutions, we would end up where we are today. Therefore, this proposed solution does not present any new ethical concerns or detriment society in any way that is not currently under threat of. Our proposed solution is an ethical measure.

7 Future Research

Edge computing is still in its early stages of exploration and more development needs to be done. Comparing the number of packets dropped, latency, and throughput compared to cloud computing still needs a lot of research. More research on what type of edge nodes are most effective in reducing latency and packet dropping and how edge computing affects energy efficiency in different systems compared to what is currently used could further propel technology. We may explore how effective edge computing would affect energy efficiency in other sectors that do not need real-time data as often [9].

In addition, it is important to note that while there have been solutions proposed to specific sectors, solutions may be applied in many scenarios unmentioned. For example, while Google DeepMind is able to effectively cool data centers, this same technique may be used to control cooling in the commercial sector. Microsoft SilviaTerra may be modified to examine coral reefs and monitor their health, as they are one of the largest producers of oxygen on the planet [16]. IIoT solutions to may be modified to run machinery in commercial factories more efficiently. MCSN's may be modified to function in vehicles to make them even more energy efficient. Solutions proposed in this paper are robust enough to be applicable to any modern technical system.

8 Conclusion

Buildings, transportation networks, and industrial systems are large and difficult to manage. AI can keep track of the many different system parameters, as outlined in Table 4, better than humans can. Currently, AI helps automate tasks, but no system has properly integrated the IoT, although future systems, shown in Fig. 6 are promising.

Table 4. Abbreviated list of parameters kept track of in each sector

Residential	Transportation	Industrial
– Air Temperature	– Gas Mileage	– Assembly Lines
– Light	– CO ₂ Output	– Lighting
– Humidity	– Avoiding Traffic	– Equipment
– Water	– Brake Wear	– Loading

Using IoT sensors that relay information about status to humans to processing nodes where AI and ML algorithms function, AI can more accurately allocate energy within a system and run systems more efficiently based on human goals.

The result has proven to be reduced energy consumption, minimized human error, lower risk to human health, improved operational efficiency, reduced costs, improved productivity, and higher quality maintenance and customer satisfaction [8].

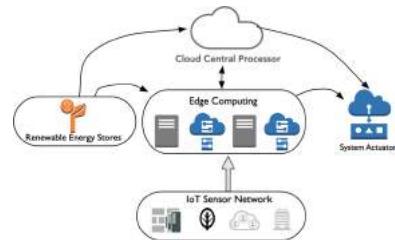


Fig. 6. Future systems that utilize AI and IoT

AI coupled with improved hardware technology can rapidly process the data of any system, which can easily be acquired by the IoT to make effective use of AI technology. By collecting specific information about the environment and presenting the results to users, humans can make decisions about their wasteful habits, can identify trends of energy mismanagement, and can correct practices damaging to the Earth.

References

1. Bogolea, B.D., Boyle, P.J., Shindyapin, A.V.: Artificial intelligence-based energy auditing, monitoring and control. WO Patent, 118,128, 18 October 2007
2. Huntingford, C., et al.: Machine learning and artificial intelligence to aid climate change research and preparedness. Environ. Res. Lett. **14**(12), 1–15 (2019)
3. Tushar, W., et al.: Internet of things for green building management: disruptive innovations through low-cost sensor technology and artificial intelligence. IEEE Signal Process. Mag. **35**(5), 100–110 (2018). <https://doi.org/10.1109/MSP.2018.2842096>
4. Mirzaei, H., Fathollahi, M., Givargis, T.: OPEB: open physical environment benchmark for artificial intelligence. In: Computing Research Repository, vol. 1707.00790, July 2017
5. Niemeyer, G., Naima, R., Garcia, A., Kaltman, E.H.: Multimodal climate sensor network. U.S. Patent, 9,332,322, 3 May 2016
6. Milano, M., O'Sullivan, B., Gavanelli, M.: Sustainable policy making: a strategic challenge for artificial intelligence. AIMag **35**(3), 22–35 (2014)
7. Kasim, H., Omar, R., Hilme, M.H.B.M., Al-Ghaili, A.M.: Future fuels for environmental sustainability: roles of computing. Int. J. Adv. Sci. Technol. **28**(10), 87–95 (2019)
8. Aazam, M., Zeadally, S., Harras, K.A.: Deploying fog computing in industrial internet of things and industry 4.0. IEEE Trans. Industr. Inf. **14**(10), 4674–4682 (2018). <https://doi.org/10.1109/TII.2018.2855198>

9. Rahman, T., Yao, X., Tao, G., Ning, H., Zhou, Z.: Efficient edge nodes reconfiguration and selection for the internet of things. *IEEE Sens. J.* **19**(12), 4672–4679 (2019). <https://doi.org/10.1109/JSEN.2019.2895119>
10. Ali, M.M.: Energy efficient architecture and building systems to address global warming. *Leadersh. Manag. Eng.* **8**(3), 113–123 (2008). [https://doi.org/10.1061/\(ASCE\)1532-6748\(2008\)8:3\(113\)](https://doi.org/10.1061/(ASCE)1532-6748(2008)8:3(113))
11. American Geosciences Institute: What are the major sources and users of energy in the United States? American Geosciences Institute. <https://www.americangeosciences.org/critical-issues/faq/what-are-major-sources-and-users-energy-united-states>. Accessed 24 June 2020
12. Evans, R., Gao, J.: DeepMind AI reduces google data centre cooling bill by 40%. DeepMind, July 2016. <https://deepmind.com/blog/article/deepmind-ai-reduces-google-data-centre-cooling-bill-40>. Accessed 18 Jan 2020
13. Hölzle, U.: Data centers get fit on efficiency. Ethical Corporation, June 2019. <https://www.ethicalcorp.com/can-ai-light-way-smarter-energy-use>. Accessed 13 Dec 2019
14. Mehta, A.: Can AI light the way to smarter energy use?. Reuters Events: Ethical Corporation, June 2016. <https://blog.google/outreach-initiatives/environment/data-centers-get-fit-on-efficiency/>. Accessed 18 Jan 2020
15. Dobrin, A.: 3 approaches to ethics: principles, outcomes and integrity. *Psychology Today*, 18 May 2012
16. Farmen, N.: How AI Is helping solve climate change. Smashing Magazine, September 2019. <https://www.smashingmagazine.com/2019/09/ai-climate-change/>. Accessed 21 Sept 2020
17. Hursthouse, R., Pettigrove, G.: Virtue Ethics. Stanford Encyclopedia of Philosophy, December 2016. <https://plato.stanford.edu/entries/ethics-virtue/>. Accessed 21 Sept 2020
18. Sinnott-Armstrong, W.: Consequentialism. Stanford Encyclopedia of Philosophy, June 2019. <https://plato.stanford.edu/entries/consequentialism/>. Accessed 21 Sept 2020
19. Alexander, L., Moore, M.: Deontological ethics. Stanford Encyclopedia of Philosophy, October 2016. <https://plato.stanford.edu/entries/ethics-deontological/>. Accessed 21 Sept 2020



Construction Site Layout Planning Using Multiple-Level Simulated Annealing

Hui Jiang¹(✉) and YaBo Miao²

¹ CTO-Office of Glodon, Beijing, China

² Technology Center of Glodon, Beijing, China

Abstract. Construction Site Layout Planning (CSLP) has always been a very challenging issue in the field of architecture technology. Generative Design that has gradually emerged in recent years provides a new perspective for solving CSLP problems. Simulated Annealing is one of the main algorithms in Generative Design, and it can quickly find the optimal solution that satisfies the conditions through multiple loop iterations. This paper attempts to use a multiple-level simulated annealing algorithm to solve a type of problem in CSLP, which is to find the optimal planning of tower cranes and material yards in the construction site. For one thing, according to the business rules of the construction site and the input drawings, this paper optimizes the definition domain of the coordinates of the tower cranes and the material yards by using the proposed buildings and roads on the drawing. For another, according to the business independence of the tower cranes and the material yards, the paper divides the algorithm into two steps for optimization rather than simultaneous calculation. The above two improvements to the simulated annealing algorithm greatly improve the efficiency of the algorithm and reduce the complexity. The results of the tests are shown, and the difficulty of selecting weights of sub-functions is also proposed at the end of the paper.

Keywords: Construction site layout planning · Simulated annealing · Artificial intelligence · Architectural layout · Generative design

1 Introduction

The issue of Construction Site Layout Planning (CSLP) has always been a hot issue in the field of AI+ Architectural Design [1]. CSLP is to make reasonable planning and layout of the main mechanical equipment, material storage yard, road traffic, temporary housing, temporary water and electricity pipelines, etc. in the construction site in accordance with the requirements of the construction plan and construction schedule. Some work of the GD (Generative Design) that has gradually emerged in recent years focuses on the solution of CSLP.

For housing construction projects, it is particularly important to determine the number and location of large-scale equipments and material storage yards, because they involve the rental of equipments and transportation cost of materials, which account for a large proportion of the all cost. This paper is trying to use the simulated annealing

algorithm to intelligently arrange the tower crane and material yard (TCL&MYL, Tower Crane Layout and Material Yard Layout).

This paper mainly applies the Multiple-level Simulated Annealing (M-SA) algorithm to explore the TCL and MYL problems in the CSLP.

2 Simulated Annealing Algorithm

The idea of simulated annealing algorithm comes from the simulation of the cooling process of solid annealing. That is to heat the solid to a sufficiently high level, and then let it slowly cool down. When the solid is heated, the thermal motion of the atoms in the solid continues to increase and the internal energy increases. With the continuous increase of temperature, the long-range order of the solid is completely destroyed, and the internal particles of the solid become disordered with the increase of temperature. When cooling, the particles gradually tend to be ordered, reaching an equilibrium state at each temperature, and finally reaching the ground state at room temperature, while the internal energy is also reduced to a minimum.

The earliest idea of simulated annealing algorithm was proposed by Metropolis in 1953. And in 1983, Kirkpatrick and others successfully introduced the annealing idea into the field of combinatorial optimization. It has been widely used in engineering, such as production scheduling, control engineering, machine learning, neural networks, image processing and other fields.

The essence of the simulated annealing algorithm lies in the Metropolis criterion, which can probabilistically jump out of the local optimal solution and eventually tend to the global optimal solution. This can be seen from the flow of the simulated annealing algorithm [2, 3]:

Suppose the objective function is $y = f(x)$, we require the minimum value of the objective function.

Step1: Select the initial control temperature T_0 , Markov chain length L_0 , randomly select an initial solution i in the feasible solution space, and the cooling function, that is, the control parameter attenuation Function T_k ;

Step2: Generate a random disturbance and get a new solution j in the feasible solution space;

Step3: Determine whether to accept the new solution, the judgment criterion is the Metropolis criterion:

(I) If $f(i) \geq f(j)$, then accept the new solution j , at this time the optimal solution $I = j$;

(II) If $f(i) < f(j)$, then accept the new solution j according to probability, namely

$$p = \exp\left(-\frac{f(j)-f(i)}{T}\right) > \text{random } [0, 1], \text{ accept the new solution } j.$$

At this time, the optimal solution $i = j$, otherwise reject j , and the optimal solution is still i ;

Step4: Repeat Step2 and step3 for L_0 times, to get an optimal solution under the Markov process with chain length L_0 ;

Step5: Determine whether the stopping criterion is met, if it is satisfied, the optimal solution will be output and the algorithm will stop, otherwise step6 will be executed;

Step6: The number of iterations $k = k + 1$, the optimal solution is updated to the solution obtained at step4, the temperature function becomes T_{k+1} , the length of the Markov chain becomes L_{k+1} , back to step2.

3 Problem Description

3.1 Input and Output of the Algorithm

The plane position of the tower crane mainly depends on the plane shape of the building and the surrounding site conditions (Fig. 1). It is generally arranged next to the building with a distance of about 5 m from the wall; the service radius of the tower crane should be basically 95% of the planned construction, and the arrangement of the group towers should be able to avoid each other from the tower body, and the coverage area should preferably be a small part of the overlap, covering the length that is 40% to 80% of the radius of the smaller tower crane [4].



Fig. 1. Tower crane in construction site

The location coordinates of the material storage yard should be placed on the side of the road as far as possible to facilitate transportation, and within the range of the tower crane, generally at least 100 m^2 as shown in Fig. 2. (The material storage yards below are uniformly arranged according to each 100 m^2).

3.2 Design Objectives and Variables

Input the Json file of the construction site drawings, you can read the outline of the entire work area, the buildings, the roads, the obstacles, etc. It is required to output the number



Fig. 2. Material yard in construction site

of tower cranes n ; the tower crane coordinates (x_i, y_i) and radius r_i , where $I = 1, 2, \dots, n$; the number of yard m , the center coordinates of the yard a_j, b_j , where $j = 1, 2, \dots, m$ (Fig. 3).

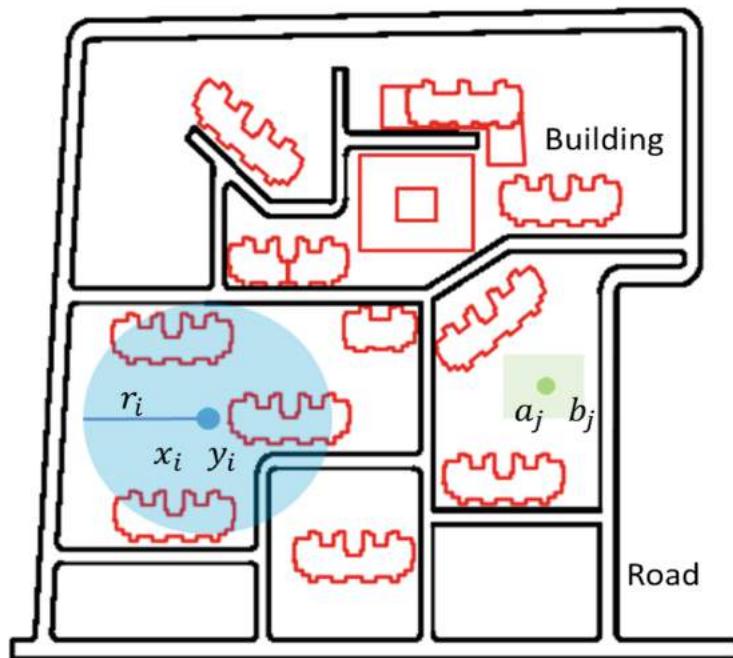


Fig. 3. Parameters in CSLP

3.3 Domain of Definition

According to the building design codes, the tower crane is required to be 3–6 m away from the outline of the proposed buildings. The domain design of the center coordinates of the tower cranes: we can simply set the value range of the tower cranes' coordinates (x_i, y_i) to the entire construction area, but this undoubtedly increases the difficulty of subsequent solutions. In order to improve the efficiency of the solution, we need to minimize the alternative range of the solution variables. We can set the entire proposed contour B in the Fig. 4 to extend 4.5 m (that is, the middle value of 3–6 m), and set it to Build_E, and its parameter is set to t . The Build_E can be regarded as $E_1(t)$, and t is the length that is a value ranging from 0 to the length of Build_E. Among them, during the external expansion process, the protruding corners need to be removed to ensure that the distance between each point on $E_1(t)$ and the proposed contour is 4.5 m as shown in Fig. 4(left).

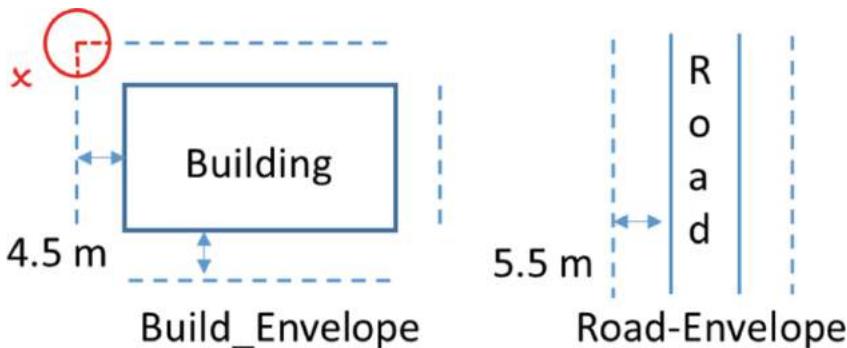


Fig. 4. Envelope design in CSLP

Similarly, we design the domain of the center coordinates of the yard.

We can set the outer contour of the road in the Fig. 4(right), and the Roads are expanded by 5.5 m (the yard is a square with a side length of 10 m, and the nearest side is 0.5 m from the edge of the road) as the Road_Envelope, and all the proposed outer contours of Envelope can be shown by the parameter s .

E_2 can be regarded as the parameter s function, i.e. $E_2(s)$, and the s is the length value, and the range of s is the length value that is from 0 to the Road_Envelope Length.

4 Objective-Functions Design

According to the business rules of TCL and MYL, objective sub-functions can be designed as follows:

- The coverage ratio of the tower cranes to the planned buildings should be as large as possible, and the coverage ratio must be greater than 95%. So the objective function f_1 is as the following formula, and the f_1 value range is [0, 1] by normalization.

$$f_1 = \frac{\text{Intersection_Area}(\bigcup_{i=1}^n C_i, B)}{\sqrt{\text{Area}(\bigcup_{j=1}^m B_j)}}$$

- The intersection of every two tower cranes cannot exceed 40%–80% of the smaller crane radius. So the objective function f_2 is as the following formula, and the f_2 value range is [0, 1] by normalization.

$$f_2 = \frac{2}{n(n-1)} \sum_{i=1}^{n-1} \sum_{j=i+1}^n \text{Bool}(0.4 < \frac{r_i + r_j - \text{Distance}(O_i, O_j)}{\min(r_i, r_j)} < 0.8)$$

- The center of the tower cranes cannot be located on the road. So the objective function f_3 is as the following formula, and the f_3 value range is [0, 1] by normalization.

$$f_3 = 1 - \frac{\text{Intersection_Area}(\bigcup_{i=1}^n C_i^q, \text{Road})}{\sqrt{\text{Area}(\bigcup_{i=1}^n C_i^q)}}$$

- The sum of tower cranes radius is required to be minimum value because the use cost of tower crane is minimal. So the objective function f_4 is as the following formula, and the f_4 value range is [0, 1] by normalization.

$$f_4 = \frac{\sum_{i=1}^n r_i - n * r_{\min}}{n * (r_{\max} - r_{\min})}$$

The above objective sub-functions 1–4 are all related to the arrangement of the tower crane. And other four objective sub-functions are related to the arrangement of the Material yard.

- The Material yard cannot be located in the proposed building, i.e., the Material yard and the proposed building cannot have an overlapping area. So the objective function f_5 is as the following formula, and the f_5 value range is [0, 1] by normalization.

$$f_5 = \frac{\sum_{j=1}^m \text{Intersection_Area}(Y_j, B)}{\sum_{j=1}^m \text{Area}(Y_j)}$$

- The material yard cannot have an overlapping area with the tower crane’s “minimum lifting range” that is the inner circle of radius 0.5m. So the objective function f_6 is as the following formula, and the f_6 value range is [0, 1] by normalization.

$$f_6 = \frac{\sum_{i=1}^n \sum_{j=1}^m \text{Intersection_Area}(C_i^p, Y_j)}{\sum_{i=1}^n \text{Area}(C_i^p)}$$

- Any two material yards cannot have an overlapping area. So the objective function f_7 is as the following formula, and the f_7 value range is [0, 1] by normalization.

$$f_7 = \frac{2}{m(m-1)} \sum_{j=1}^{m-1} \sum_{j=i}^m \frac{\text{Intersection_Area}(Y_i, Y_j)}{\min(\text{Area}(Y_i), \text{Area}(Y_j))}$$

8. Each tower crane must service one material yard at least, and the overlapping area must cover the material yard above 80%. So the objective function f_8 is as the following formula, and the f_8 value range is [0, 1] by normalization.

$$f_8 = \frac{1}{n} \sum_{i=1}^n \text{Bool}\left(\frac{\text{Intersection_Area}(C_i, \bigcup_{j=1}^m Y_j)}{\frac{1}{m} \sum_{j=1}^m \text{Area}(Y_j)} > 0.8\right)$$

To sum up, we can get the total objective function [5, 6]:

$$f_{all} = w_1 * f_1 + w_2 * f_2 + w_3 * f_3 + w_4 * f_4 + w_5 * f_5 + w_6 * f_6 + w_7 * f_7 + w_8 * f_8$$

Where w_i is a function operator, which can be multiplication, reciprocal, square, cube and other operations.

We use python3.7 version to implement the above scheme, and then we discuss the results.

5 Results

For the objective function formed by the above 8 sub-functions, the value of the operator w_i is difficult to determine.

Considering two main types of optimization objectives: tower cranes and storage yards, multiple level simulated annealing algorithms are used: Mutiple-SA.

The user need input the site drawing in json format including the proposed buildings outline and roads outline as shown in Fig. 5; the user enters the optional value of the tower crane arm length, such as 50 m or 60 m; the user enters the required material yard area, for example, 400 m².

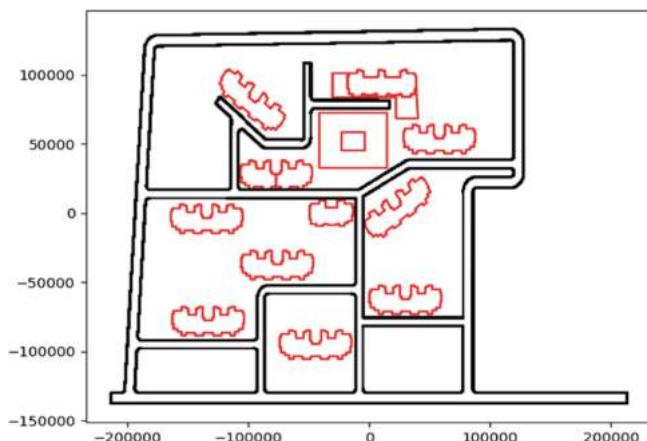


Fig. 5. JSON File of drawings export

Algorithm Steps

1. First of all, the central value of the number of tower cranes is obtained by the hierarchical clustering algorithm, but it is time-consuming. In order to improve the calculation efficiency, we adopt the estimation method, i.e., according to the test case drawings, the proposed buildings are nearly evenly distributed in the construction site, so the estimated number of tower cranes n_0 can be obtained that it is equal to the area of the construction site area is dived by area of the largest crane boom.
2. According to the actual measurement of the test case drawings the number of tower cranes is n , and the value range can be set to $[n_0 - 2, 2 * n_0]$.
3. Set each material yard to be a square with a side length of 10 m. The number of material storage yards is the total area of the material yard required by the user divided by 100, which is set as m ;
4. Import the number of tower cranes n and the number of material yard m into the MSA algorithm as variables.

For the first level SA, the objective function: $\text{Object_F}_1 = w_1 * f_1 + w_2 * f_2 + w_3 * f_3 + w_4 * f_4$, where the operator w_i is defined as multiplication, and the values could be 100000, 1000, 100, 100.

Through this step, we can get the coordinates and radius of the tower cranes (see Fig. 6).

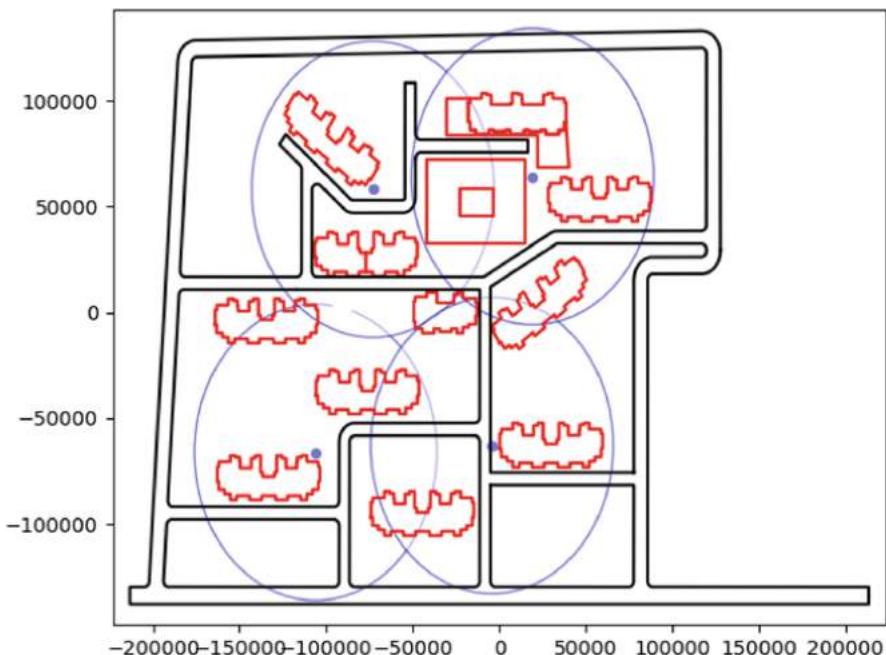


Fig. 6. TCL test-results using $\text{object_F}_1 = w_1 * f_1 + w_2 * f_2 + w_3 * f_3 + w_4 * f_4$

5. In the second level SA, the objective function Object_F₂ = w₅f₅ + w₆f₆ + w₇f₇ + w₈f₈. We can obtain the center coordinates of the material yard with keeping the tower crane coordinates and radius unchanged.

From the Fig. 7, it can be seen that this kind of layout Planning has a better effect.

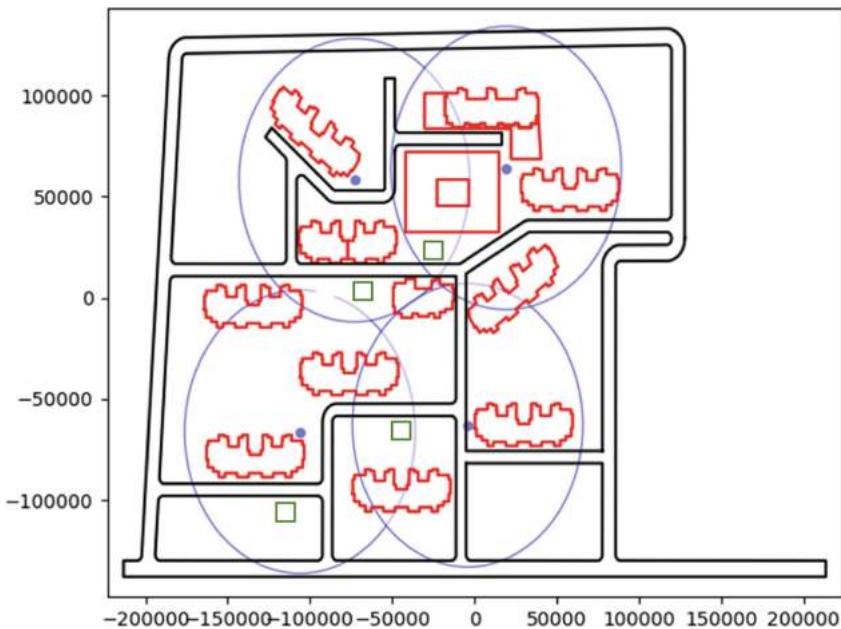


Fig. 7. MYL test-result using object_F₂ = w₅f₅ + w₆f₆ + w₇f₇ + w₈f₈

6 Conclusions

Based on the simulated annealing algorithm, we can indeed solve some of the CSLP problem. However, because of the large number of CSLP business needs, there are still many business issues have not been met. In order to solve these issues, it is necessary to increase the objective function, but it also increases the difficulty of selecting the weight of each sub-function. In the future, the choice of these weights may be calculated by some new algorithms.

References

1. Kumar, S.S., Cheng, J.C.P.: A BIM-based automated site layout planning framework for congested construction sites. *Autom. Constr.* **59**, 24–37 (2015)
2. Chen, H., Wu, J., Wang, J.: Mechanism study of simulated annealing algorithm. *J. Tongji Univ. (Nat. Sci.)* **32**(6), 802–805 (2004)

3. Hao, Z., Yue, R.: Architecture layout design through simulated annealing algorithm. In: 25th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), vol. 1, pp. 275–284 (2020)
4. Zhu, L., Lu, W.: Analysis of construction site tower crane layout essentials. *Constr. Technol.* **47**(12), 1746–1748 (2018)
5. Hea, S., Perret, J.: A stochastic method for the generation of optimized building layouts respecting urban regulations. <https://www.academia.edu/23918866/>
6. Zawidzkia, M., Tateyamaa, K.: The constraints satisfaction problem approach in the design of an architectural functional layout. <https://www.tandfonline.com/doi/abs/10.1080/0305215X.2010.527005>



Epistocracy Algorithm: A Novel Hyper-heuristic Optimization Strategy for Solving Complex Optimization Problems

Seyed Ziae Mousavi Mojab¹(✉), Seyedmohammad Shams², Hamid Soltanian-Zadeh², and Farshad Fotouhi¹

¹ Department of Computer Science, Wayne State University, Detroit, MI 48202, USA
mousavi@wayne.edu

² Department of Radiology, Henry Ford Health System, Detroit, MI 48202, USA

Abstract. This paper proposes a novel evolutionary algorithm called Epistocracy which incorporates human socio-political behavior and intelligence to solve complex optimization problems. The inspiration of the Epistocracy algorithm originates from a political regime where educated people have more voting power than the uneducated or less educated. The algorithm is a self-adaptive, and multi-population optimizer in which the evolution process takes place in parallel for many populations led by a council of leaders. To avoid stagnation in poor local optima and to prevent a premature convergence, the algorithm employs multiple mechanisms such as dynamic and adaptive leadership based on gravitational force, dynamic population allocation and diversification, variance-based step-size determination, and regression-based leadership adjustment. The algorithm uses a stratified sampling method called Latin Hypercube Sampling (LHS) to distribute the initial population more evenly for exploration of the search space and exploitation of the accumulated knowledge. To investigate the performance and evaluate the reliability of the algorithm, we have used a set of multimodal benchmark functions, and then applied the algorithm to the MNIST dataset to further verify the accuracy, scalability, and robustness of the algorithm. Experimental results show that the Epistocracy algorithm outperforms the tested state-of-the-art evolutionary and swarm intelligence algorithms in terms of performance, precision, and convergence.

Keywords: Epistocracy algorithm · Evolutionary computation · Metaheuristic optimization algorithm · Multi-dimensional search · Swarm intelligence

1 Introduction

Evolutionary computation (EC) is a subfield of artificial intelligence that encompasses methods mimicking mechanisms of biological evolution to solve various optimization problems. An optimization problem essentially requires finding a set of parameters $\vec{x} = (x_1, \dots, x_n) \in S$ of the current system, such that a certain quantity $f : S \rightarrow \mathbb{R}$ is maximized (or minimized) $\forall \vec{x} \in S : f(\vec{x}) \leq f(\vec{x}^*)$.

Over the past few decades, many state-of-the-art evolutionary algorithms such as Genetic algorithm (GA) and Evolutionary Strategies (ES) have been proposed for applications where a well-defined or closed-form solution does not exist [1]. Genetic algorithm was developed by John Holland in the early 1970s [2–4] mimicking Darwinian theory of survival of the fittest and Evolutionary Strategies founded by Rechenberg and Schwefel in 1965 [5–7] based on the hypothesis that small mutations occur more commonly than large mutations. Both Genetic algorithm and Evolutionary Strategies rely on the concept of population, representing potential solutions to the optimization problem which iteratively undergo genetic operators to improve their fitness score. While Genetic algorithms use a binary string of digits to represent solutions and use both mutation and recombination as genetic operators, in Evolutionary Strategies a fixed-length real-valued vector is used for representation, and only mutation is used as a primary search operator. In evolutionary algorithms, the recombination operator performs an information exchange, and the mutation operator generates variations of the solutions and increases the diversity among the population. The selection operator, however, makes better individuals to survive and reproduce.

Another subset of nature-inspired algorithms is Swarm Intelligence (SI) which is based on collective behavior of a decentralized, self-organizing network of agents such as bird flocks or honeybees. In SI algorithms, multiple agents can locally interact and exchange heuristic information which leads to the emergence of global behavior of adaptive search and optimization. Particle Swarm Optimization (PSO) is an example of swarm intelligence proposed by Eberhart and Kennedy in 1995 [8]. This algorithm is inspired by social behavior of bird flocking and fish schooling. Similar to GA, PSO is initialized with a population of random candidate solutions that are improved iteratively over time, however, unlike GA has no evolution operators such as recombination and mutation. Despite the fact that PSO is a powerful and effective optimization technique, it still suffers from stagnation and premature convergence [9, 10]. Several solutions including inertia weight, and time-varying coefficients have been proposed to eliminate these problems [11, 12].

The Artificial Bee Colony (ABC) is another popular swarm intelligence-based algorithm which is inspired by the foraging behavior of the honeybees. ABC consists of three groups of bees: employed bees, onlookers, and scouts that have different roles in the optimization process. ABC is simple, easy to implement, and highly flexible [13]. This algorithm was first proposed by Dervis Karaboga in 2005 [14] to optimize numerical problems. Since then, many variants of ABC have been introduced to increase the population diversity and avoid premature convergence [15, 16].

Cuckoo Search Algorithm (CSA) is one of the latest swarm intelligence-based algorithms developed by Yang and Deb in 2009 [17]. This algorithm is inspired by natural behavior of cuckoos who lay their eggs in other birds' nests for breeding. Compared to other approaches, Cuckoo requires fewer numbers of parameters to be fine-tuned. In 2018, Mareli *et al.* [18] developed three new Cuckoo search algorithms using linear, exponential and power increasing switching parameters to maintain an optimum balance between local and global exploration and increase the efficiency of CS algorithm. In 2019, Li *et al.* [19] proposed a new variant of CSA called I-PKL-CS algorithm which employs self-adaptive knowledge learning strategies to mitigate premature convergence

and poor balance between exploitation and exploration. I-PKL-CS exploits individual and population knowledge learning to improve the quality of solutions and convergence rate.

There exist many real-world applications for EC. In [20], the genetic algorithm was used to decrease the dimension of the data and to optimize the weights and biases of the neural network in ECG signal classification. Xi *et al.* [21] used PSO to improve the performance of their neural network in order to assess the hazard of earthquake-induced landslide. Kim *et al.* [22] used self-adaptive Evolutionary Strategies to optimize the parameters of an autonomous car controller. Prakash *et al.* [23] employed the Cuckoo Search algorithm to perform job scheduling and resource allocation on the grid. Yeh *et al.* [24] used ABC to optimize a bee recurrent neural network to generate a novel approximate model for predicting network reliability.

The selection of an evolutionary approach can drastically reduce the amount of time needed for finding an optimal solution. According to several studies, evolutionary algorithms, in general, suffer from various problems such as limited searching ability [25–27], curse of dimensionality and scalability [28, 29], premature convergence and stagnation [30–33], and poor performance which usually occur in the absence of population diversity and adaptability [34–36], and due to unbalanced exploration-exploitation capacities [37, 38].

The work reported in this paper was motivated by the fact that optimization algorithms require new explorative and exploitative capabilities along with a dynamic resource allocation technique and diversification strategies to help them converge to the optimal solution at the early stages of the optimization process. There is a need for a new generation of evolutionary algorithms that can avoid entrapment in local optima and prevent premature convergence [30, 39]. To find the optimal solution, these algorithms must employ a directed and goal-oriented search rather than a purely random and stochastic one.

In this paper, we propose a new hyper-heuristic algorithm based on a political regime called Epistocracy where educated people have more voting power (weight) than the uneducated or less educated. The Epistocracy algorithm splits the population into Governors and Citizens based on the performance of the individuals. The Citizens are assigned Governors based on the degree of similarity and the exercise of free will. Once a Citizen is assigned a Governor, they move towards their Governor in an attempt to mimic some of the traits which made their Governor successful. Governors will also try to improve themselves and lead their population to collaboratively search for the optimal solution.

The Epistocracy algorithm is a self-adaptive, and multi-population optimizer in which the evolution process takes place in parallel for many populations led by a council of leaders. To avoid entrapment in poor local optima and to prevent a premature convergence, the algorithm employs multiple mechanisms such as dynamic and adaptive leadership based on gravitational force, dynamic population allocation and diversification, variance-based step-size determination, and regression-based leadership adjustment. The algorithm uses a stratified sampling method called Latin Hypercube Sampling (LHS) to distribute the initial population more evenly for exploration of the search space and exploitation of the accumulated knowledge.

The rest of the paper is organized as follows. Section 2 describes the overall structure of the Epistocracy algorithm in detail. Experimental results and comparative studies on benchmark test functions along with Convolutional Neural Networks (CNNs) parameter optimization are presented in Sect. 3. Finally, conclusions and directions for future research are presented in Sect. 4.

2 Epistocracy Algorithm

2.1 Overview

The term Epistocracy is derived from the Greek word *epistêmê* meaning knowledge, knowing, and understanding. John Stuart Mill (1806–1873), the British philosopher and political economist in his book “Mill on Bentham and Coleridge” proposed to give more votes to the better educated [40]. Jason Brennan believes that more competent or knowledgeable citizens must have slightly more political power than less competent citizens [41]. In fact, the problem with democracy is the elimination of the epistemic dimension of democracy. While Democracy is more about the input aspect of the decision-making process, Epistocracy is concerned about the output.

Epistocracy algorithm is multi-population optimization algorithm which seeks to minimize the time taken to find an optimal value for the problem being solved. As an adaptive, hyper-heuristic algorithm, Epistocracy employs problem-related knowledge, and globally aggregated statistics to automatically adjust itself during each run and search through a space of meta-heuristics to find the optimal solution. Epistocracy attempts to incorporate human sociopolitical behavior and intelligence to improve the performance and convergence speed and reduce the probability of getting trapped in local optima compared to other meta-heuristic algorithms.

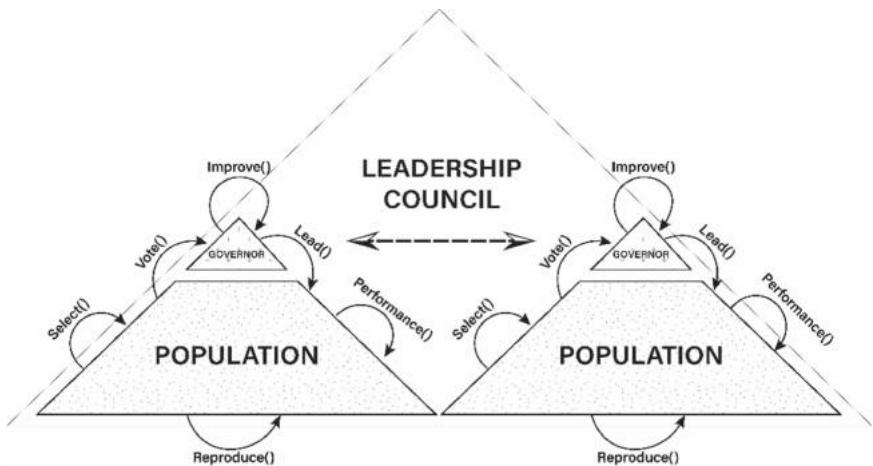


Fig. 1. Flow diagram of epistocracy algorithm.

As illustrated in Fig. 1, the Epistocracy algorithm is made of two primary components: Governors and Citizens. Citizens are individual solutions that are randomly

and uniformly created. In each generation, all individuals are evaluated with a pre-defined fitness function. The top-performing individuals (Governors) are then selected through the Select() function to lead the population. Governors are, in fact, a network of cooperative leaders who influence and evolve the generation of the new population via Lead() function. While Governors continuously improve themselves, citizens can vote for governors and affect their position in the government. Information is systematically propagated among citizens and governors. Figure 2 shows the flowchart of the proposed algorithm.

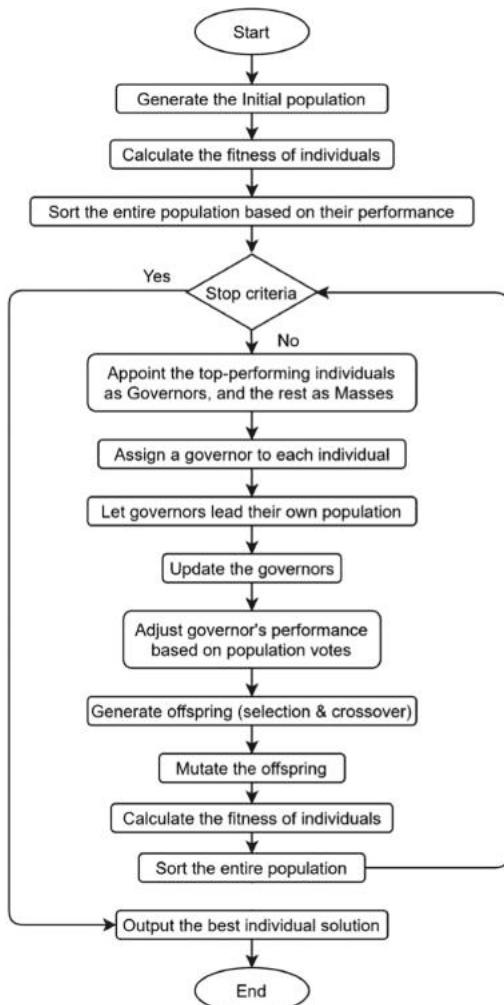


Fig. 2. Flowchart of the epistocracy algorithm with all steps involved from the population generation until outputting the optimal solution.

2.2 Generating the Initial Population

The Epistocracy algorithm starts the optimization process by generating a population of random solutions, using a stratified sampling method called Latin Hypercube Sampling (LHS) which was originally proposed by McKay in 1979 [42]. Each individual solution has a set of genes or attributes known as chromosome which are defined using their corresponding upper and lower bounds in the search space. In this algorithm, the set of attributes represent the initial position and level of political knowledge of each individual in the society.

2.3 Performance Evaluation

The performance of an individual in the population is evaluated using a pre-defined fitness function. Given the individual's current chromosome, the actual performance is calculated and stored as an individual's "actual performance". The previous actual performance is also recorded for future reference.

Individual solutions are then ranked based on their actual performance (fitness score). The adjusted performance is calculated based on the actual performance of each individual solution. The calculation steps will be explained in detail in the following sections.

2.4 Population Separation

Different people demonstrate different understandings of patterns of change and achieve different levels of success and result upon the social hierarchy. This algorithm plans to separate the Governors from Citizens based on the level of success an individual can achieve. The top performers in the population will be considered "Governors" and the rest will be considered "Citizens".

2.5 Governor Assignment

Before evolving each individual and moving them around, about five percent of the top-performing individuals in each generation are selected as a set of governors to lead the population and help them improve their performance. Transcending traditional societies, in Epistocracy, governments have no obvious borders, and individuals can follow or vote for any governor anywhere expressing the idea of "Global Village". In the Epistocracy algorithm, each individual is assigned to a governor based on their phenotypic characteristics, and the degree of influence and impact of the governor on the citizen. To that end, the Gravitational Force (1) is used to calculate the magnitude of attraction between each citizen and every governor. A governor with a larger gravitational force has a higher probability to attract a citizen and form a larger territory. However, some citizens may act as rebels and resist against the orders of the befitting authorities and may follow different governors.

$$F = G \times \left(\frac{m_1 \times m_2}{r^2} \right) \quad (1)$$

In the above equation of the Gravitational Force, G is a constant, and m_1 , and m_2 , are the adjusted performances of the governor and citizen respectively. All performances are normalized using the following formula:

$$P_{norm\ i} = \left[\frac{P_i - \min(P_{governors})}{\max(P_{governors}) - \min(P_{governors})} \right]^{-1} \quad (2)$$

In (2), P_i is the individual's actual performance, where $P_{governors}$ is the list of governors' performances.

The Euclidean distance (3) is used to calculate the distance r between a governor and a citizen.

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \|\mathbf{x}_i - \mathbf{x}_j\| = \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \quad (3)$$

$n = \text{number of variables}$

To imitate the rebelliousness of citizens, a roulette wheel with the governors' calculated gravitational forces is used to give citizens a freedom of selecting other governors with even a greater dissimilarity (distance). This will help the algorithm to explore the inter-space between the governors by moving a citizen across the governments. The selection probability is defined using the following equation:

$$P_j = \frac{G(S_j)}{\sum_{i=1}^n G(S_i)} \quad (4)$$

In (4), n is the number of governors. G is the gravitational force of solution S_j .

In the next generation, if the assigned governor is overthrown or resigned due to their poor performance or their own population votes, the surviving citizen will choose a new governor from the updated list of governors. If a governor performs poorly, eventually, he will be degraded and may lose all his population and get removed from the current list of governors. This happens when a population's total performance over a certain period of time (an iteration) compared to other populations is very small. In this case, the governor will lose his popularity regardless of his own performance at the time of being selected. In fact, a governor's popularity rests on his credibility and competence, and his performance in leading his population and improving their lives. By adjusting the actual performance of the governor, the governor's rank in the governors list will change. Given that, this governor will have a lower chance to be selected by new citizens who do not have any governor yet.

2.6 Leading the Population

In the next step, the Epistocracy algorithm allows governors to lead their own population. Each citizen will take a step of variable length (5) toward his governor to improve his performance and become similar and even better than their governor. The step size is proportional to the distance between the governor and citizen and inversely proportional to the self-improvement of the citizen under the rule of the governor. The following formula is used to calculate the next step of each citizen:

$$S_i = \left(\frac{I_{avg}}{I_{min}} \right) \times \sigma^2 \times d_{i,g} \times \varphi \quad (5)$$

where S_i is the individual's new step size, and I_{avg} is the average improvement of the governor's sub-population (7). I_{min} is the minimum improvement in the population. σ^2 is the variance of the sub-population, and d_{ig} is the Euclidean distance between the individual and its designated governor. φ is the rate of change equal to 0.1. The self-improvement is calculated as follows:

$$I_i = (P_{old\ i} - P_{actual\ i}) \quad (6)$$

The self-improvement is the difference between the old and the current actual performance of the citizen. The average improvement is then calculated by:

$$I_{avg} = \frac{1}{n} \times \sum_{i=1}^n I_i \quad (7)$$

In (7), n is the size of the governor's sub-population. The average improvement is an important factor for the step size determination. To avoid missing any minima or maxima, a smaller step will be taken when a larger improvement is achieved, and a larger step will be taken when a smaller improvement is obtained.

The population variance is given by the following formula:

$$\sigma^2 = \frac{\sum (x_i - \mu)^2}{n} \quad (8)$$

To reflect the diversity of the society, if a citizen by taking a new step becomes exactly similar to his governor or another citizen in the same population, the citizen will be mutated to save the system resources. This also helps the algorithm to avoid division by zero in calculating the gravitational force when the distance between a citizen and his governor becomes zero.

2.7 Improving Governors

Similar to citizens, governors will also improve themselves by taking a step in a direction that hopefully increases their performance. To that end, the variance of governors' population is calculated, helping governors converge toward a location with the highest possibility of finding the optimal solution. The next step size of the governor is calculated like that of a citizen. However, instead of calculating the distance between the governor and citizen, this time the governor's previous step is considered according to the following formula:

$$S_j = \left(\frac{I_{avg}}{I_j} \right) \times \sigma^2 \times S_{prev\ j} \quad (9)$$

where "previous step" S_{prev} is initialized as:

$$prevStep = (upper_{limit} - lower_{limit}) \times space_{resolution} \quad (10)$$

In (10), $upper_{limit}$ and $lower_{limit}$ are the boundaries of the search space, and the space resolution is initially set to 0.001.

Since the governor is in charge of leading his population and pushing them to the right direction, the algorithm will let the governor take a step only if that step improves

his overall performance, otherwise, the governor will stay in his previous place without making any movement.

Since for computing the new step the variance of all governors is used, in the next iterations for the same governor the step size might be different and might help the governor to get improved and consequently positively contribute to the improvement of his population. The following piecewise function (11) shows the conditional step that must be taken by each governor, provided that the step improves the governor's actual performance:

$$S_j = \begin{cases} \left(\frac{I_{avg}}{I_j} \right) \times \sigma^2 \times S_{prev,j} & \text{if } \Delta P \leq 0 \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

where $\Delta P = P_{actual\ new} - P_{actual\ old}$. This formula is designed for a minimization optimization problem.

2.8 Governor's Performance Adjustment

When a population performs well or poorly under a leadership of a governor, the algorithm will adjust the governor's actual performance to allocate the right amount of resources (individuals) to the governor. For example, if a population is performing well, that must increase the trust of the population in the governor. In this case, generally more individuals will be following the governor to help him accomplish the task of finding the optimal solution. If a governor is performing poorly, the governor's actual performance will be lowered accordingly, and eventually, some individuals will leave the governor and follow another governor to improve their quality of life. In other words, like the Epistocratic societies, when a population is under-performing, this will eventually affect the popularity and credibility of the governor. Those people who initially voted for that governor, will shift away from the governor, and try to choose another governor. In each iteration, the population will vote on the performance of the governor, however, these votes have different weights.

The Epistocracy algorithm will compute the average improvement per each population, giving higher weights to individuals who are closer to the governor (and more educated) and lower weights to citizens who are farther away (and less educated) before using the following formula:

$$I_{avg} = \frac{1}{\sum_{i=1}^n w_i} \times \sum_{i=1}^n w_i I_i \quad (12)$$

In (12), n is the size of the sub-population and I_i is the individual's self-improvement given by:

$$I_i = (P_{old_{actual,i}} - P_{actual,i}) \quad (13)$$

In (13), P_i is the individual's performance. The weight of an individual's vote, w_i is calculated as follows:

$$w_i = -\log \frac{d_{i,g}}{\sum_{k=0}^n d_{k,g}} \times \frac{P_{actual,i}}{(P_{actual,g} - P_{actual,i}) + \varepsilon} \quad (14)$$

where $d_{i,g}$ is the Euclidean distance between the individual and their governor. $\sum_{k=0}^n d_{k,g}$ is the total distance between a governor and every individual in their sub-population. P is the performance, and ε is a very small positive number. In (14), the log scale is used to mitigate the impact of extreme changes in distance calculation.

In the next step, a linear regression is used to compute the adjusted performance of each governor based on their population average performance and votes. Given I_{avg} and actual performance of each governor we calculate the predicted performance, $P_{predicted}$ as follows:

$$P_{predicted} = \beta_0 + \beta_1 \times I_{avg} + \varepsilon_i \quad (15)$$

where ε_i is the residual error whose distribution is $N(0, \lambda)$, and b_0 and b_1 are calculated as follows:

$$b_1 = \frac{\sum(I_{avg} - \bar{I}_{avg}) \times (P_{actual} - \bar{P}_{actual})}{\sum(I_{avg} - \bar{I}_{avg})^2} \quad (16)$$

$$b_0 = \bar{P}_{actual} - b_1 \times \bar{I}_{avg} \quad (17)$$

The adjusted performance, $P_{adjusted}$ is calculated using the following formula:

$$P_{adjusted} = P_{actual} + [\eta \times \Delta P] \quad (18)$$

where

$$\Delta P = P_{actual} - P_{predicted} \quad (19)$$

$$\eta = \frac{1}{n} \times \frac{s_j}{\sum_{i=1}^n s_i} \quad (20)$$

In (20), n is the number of governors, and s_j is the population size of the j th governor.

2.9 Genetic Operators: Recombination and Mutation

Finally, the genetic operators are used to generate offspring based on the initial population. To maintain genetic diversity, recombination and then mutation is applied to the existing solutions. Selection, crossover, and mutation are the three operators used in the Epistocracy algorithm. The crossover operator uses the tournament method to choose the best parents among the sampled candidates. Their chromosomes are split at randomly picked points between 1 and the chromosome size - 1. The chromosome of each of the two new offspring is made of the genes from the opposite side of the split point of each of the two parents.

A percentage of the new individuals are then mutated. Once a chromosome is selected to be mutated one of its genes is selected at random. This gene will be replaced with a random number between the upper and lower bounds. The chromosome is then validated to ensure all genes are within the bounds. If a gene is not within the bounds it will be set to the closest bound.

3 Experimental Results and Analysis

3.1 Evaluation of Epistocracy Algorithm Using Benchmark Functions

To test the performance of our proposed algorithm, we have used several multimodal benchmark functions (i.e. Eggholder, Rastrigin, Schaffer-4, CrossInTray, Griewank) with a large number of local optima from the global optimization literature. We have also used 5 state-of-the-art evolutionary algorithms to compare the consistency and reliability of the Epistocracy algorithm using these benchmark functions.

In order to make a fair comparison between Epistocracy and other state-of-the-art algorithms, we have selected a set of optimization problems, and tested each algorithm with a population size of 100, for 100 runs and 100 iterations in each run.

The results of comparison among Epistocracy, Genetic Algorithm, Evolutionary Strategies, Artificial Bee Colony, Cuckoo Search, and Particle Swarm Optimization on different functions are given in Table 1, where “Mean” indicates the average fitness obtained from 100 runs and “Std.” is the standard deviation. “Min” and “Max” are the best and worst fitness values, found throughout 100 runs, respectively.

As shown in Table 1, the Epistocracy algorithm demonstrates higher reliability and consistency compared to other algorithms due to lower variation and dispersion in the outcome of the objective function.

As illustrated in Fig. 3, the absence of outliers and smaller standard deviation represented by a tinier boxplot are the most significant advantages of the Epistocracy algorithm. According to the test results of Rastrigin 5D, Epistocracy obtained the smallest standard deviation and produced better mean than other algorithms. This is a proof that the Epistocracy algorithm can effectively avoid being trapped in local minima in a complex, multimodal environment. This also shows that our algorithm is scalable and has a clear advantage over other evolutionary algorithms tested in this problem.

For Schaffer-4 2D, Epistocracy algorithm shows a higher reliability than other algorithms by producing results within a narrower range depicted in its corresponding boxplot. With CrossInTray 2D, the Epistocracy algorithm still is either doing better than other algorithms such as GA, and ES, or performing the same as PSO. However, overall, the Epistocracy algorithm has a better consistency and reliability than PSO, and similar algorithms. Among all other algorithms, for Griewank 2D, the Epistocracy algorithm has produced a narrower range of optimal solutions which is represented by its tiny boxplot. For Griewank 5D, the Epistocracy algorithm, again, shows a better result than other algorithms, and reconfirms the reliability and consistency of the algorithm working in different environments with different characteristics.

These preliminary results also show that the Epistocracy algorithm is more reliable with functions that contain multiple minima. From the robustness aspect, the Epistocracy algorithm is more robust with respect to the existence of multiple minima. For large scale search space, the Epistocracy algorithm performs more efficiently than other algorithms. In terms of convergence, the Epistocracy algorithm showed a decent rate of convergence compared to other algorithms.

Table 1. Comparison of the benchmark functions.

Function		Epistocracy	GA	ES	ABC	CSA	PSO
Eggholder 2D	Min	-959.6407	-959.6407	-959.6407	-959.6407	-959.6407	-959.6407
	Max	-957.7592	-894.4704	-893.6453	-951.0668	-753.0372	-786.5260
	Mean	-959.5399	-938.0387	-946.7461	-958.8248	-917.5035	-927.3717
	Std.	0.4198	16.6918	22.8390	1.9929	48.1905	49.8634
Rastrigin 5D	Min	7.1054E-15	0.7325	4.1369	1.5228E-05	0.8285	1.9903
	Max	7.8160E-14	7.0667	11.5452	0.0033	5.6601	14.9245
	Mean	2.7001E-14	3.2081	8.5620	0.0012	2.5568	6.4700
	Std.	1.5046E-14	1.6610	1.9900	0.0009	1.1924	3.1550
Schaffer-4 2D	Min	0.2926	0.2926	0.2926	0.2926	0.2926	0.2926
	Max	0.2926	0.3067	0.2956	0.2929	0.3037	0.2926
	Mean	0.2926	0.2965	0.2937	0.2927	0.2957	0.2926
	Std.	8.0929E-09	0.0035	0.0009	0.0001	0.0036	1.4528E-07
CrossInTray 2D	Min	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626	-2.0626
	Max	-2.0626	-2.0620	-2.0623	-2.0626	-2.0626	-2.0626
	Mean	-2.0626	-2.0625	-2.0625	-2.0626	-2.0626	-2.0626
	Std.	9.1125E-16	0.0002	7.5410E-05	8.7118E-09	4.2811E-08	9.1125E-16
Griewank 2D	Min	0.0000	0.0082	0.0133	2.6017E-07	5.1834E-09	0.0000
	Max	0.0101	0.1353	0.1556	0.0123	0.1320	0.0099
	Mean	0.0050	0.0561	0.0728	0.0036	0.0374	0.0035
	Std.	0.0037	0.0317	0.0399	0.0046	0.0370	0.0039
Griewank 5D	Min	4.9108E-11	0.2869	0.3320	9.1378E-06	0.0652	0.0246
	Max	0.0862	1.0702	1.4168	0.0639	0.5257	0.2463
	Mean	0.0398	0.7734	1.0838	0.0309	0.2215	0.1184
	Std.	0.0190	0.2264	0.2701	0.0188	0.1217	0.0609

3.2 Evaluation of Epistocracy Algorithm Using the MNIST Dataset

To further evaluate the performance of our method, we tasked the Epistocracy algorithm to find the optimal set of hyper-parameters to build the best CNN model for “MNIST” handwritten digit recognition.

The MNIST dataset is a set of hand-written digit images ranging from 0–9. This dataset contains size-normalized, gray-scale examples of digits written by 500 writers that were centered in a 28×28 image and associated with a label from 10 classes. MNIST consists of a training set of 60,000 examples, and a test set of 10,000 examples and was constructed from NIST’s (the US National Institute of Standards and Technology) Special Database 3 and Special Database 1 which contain binary images.

Each feature vector (row in the feature matrix) consists of 784 pixels (intensities) flattened from the original 28×28 pixels images. The end goal is to classify the handwritten digits based on a 28×28 black and white image. MNIST dataset is commonly used for training classification algorithms and benchmarking purpose.

Optimization of Hyper-parameters. The problem of finding the optimal value for hyper-parameter λ is called hyper-parameter optimization. The main technique for finding such a value is to choose a value λ_i from the trial set $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$, to evaluate the response function $\Psi(\lambda)$ for each one, and return the λ_i that worked the best as $\hat{\lambda}$.

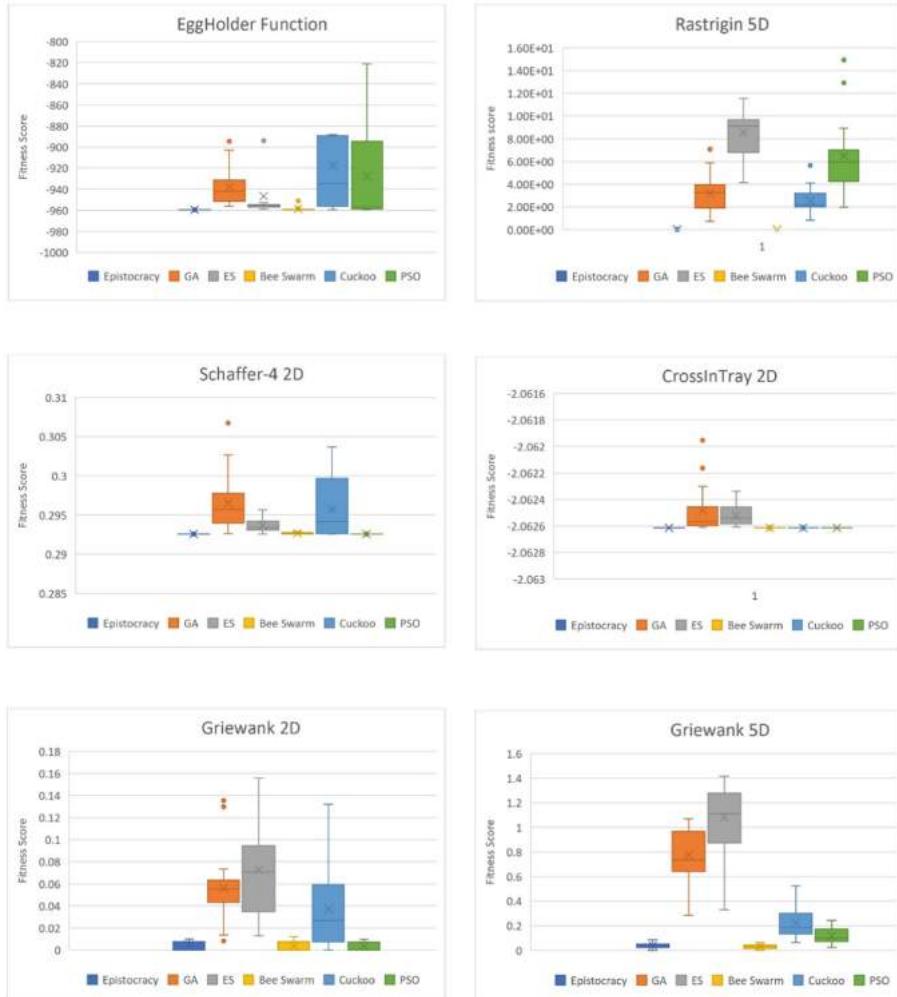


Fig. 3. Box and whisker plot of fitness scores for different benchmark functions.

The optimization of hyper-parameters can be expressed as follows:

$$\begin{aligned} \hat{\lambda} &\approx \underset{\lambda \in \Lambda}{\operatorname{argmin}} \mathbb{E}_{\sim \mathcal{G}_x} [\mathcal{L}(x; \mathcal{A}_\lambda(X^{train}))] \\ &\equiv \underset{\lambda \in \Lambda}{\operatorname{argmin}} \Psi(\lambda) \end{aligned} \quad (21)$$

In the above formula, λ is the hyper-parameter that should be selected in a way that the generalization error (loss function) $\mathbb{E}_{\sim \mathcal{G}_x} [\mathcal{L}(x; \mathcal{A}_\lambda(X^{train}))]$ minimized. \mathcal{A} is the learning algorithm that maps the training dataset X^{train} from a natural distribution \mathcal{G}_x to the function $f, f = \mathcal{A}_\lambda(X^{train})$. The hyper-parameter optimization can be denoted as the minimization of the response function $\Psi(\lambda)$ over $\lambda \in \Lambda$ where Λ is the search space.

MNIST CNNs as a Proof of Concept. Epistocracy as a multivariate optimization algorithm can be adapted for use in the automated discovery of CNN architectures, however, its effectiveness in doing so would be difficult to test. A regular multivariate optimization problem might have a known minimum or maximum while the accuracy of a CNN does not. In addition, the exact answer of most problems can be obtained through mathematical proof or exhaustive search. A full exhaustive search, however, is both time-consuming and computationally expensive, and there is no way to know what the best possible architecture of a model is.

The solution to this problem is to create a finite set of architectures and task Epistocracy with finding the best architecture in that set. For this purpose, a set of 480 unique models were generated, using all possible values of the hyper-parameters shown in Table 2. Every permutation of these hyper-parameters was used to create a distinct model. More hyper-parameters were not used since each model takes a considerable amount of time to train and test. Adding more options would make the amount of time needed to create all permutations of models unreasonable, and we must know the accuracy of all permutations in order to use MNIST as a proof of concept.

Table 2. Hyper-parameters and values used in an exhaustive search.

Hyper-parameter	Values used
Filter number	12, 16, 20, 24, 28, 32
Filter size	3, 4, 5, 6, 7
Neuron size	50, 100, 150, 200
Dropout rate	0.1, 0.2, 0.3, 0.4

Creating CNNs for MNIST. The first step to making the 480 different architectures is to create every possible set of hyper-parameters. Each set is a unique set of hyper-parameters for a single model. Given a set of hyper-parameters, a 16-layer equivalent CNN model is created in Keras using Google’s sample code to train an “MNIST” handwritten digit recognition model.

Table 3. Best combination of hyper-parameters.

Hyper-parameter	Value
Filter number	28
Filter size	6
Dropout rate	3
Neuron size	50

Once all 16 layers are created, the model is compiled with the “Adam” optimizer and “Categorical Cross-Entropy” loss function. The model is then trained with all 60,000

images in the MNIST training dataset. Each model was tested on the MNIST test set which consists of 10,000 images. After evaluating all 480 architectures the best combination of hyper-parameters was identified (see Table 3). The accuracy of this model was 99.51%.

Epistocracy was then used to search and find the same optimal set of hyper-parameters shown in Table 3. Epistocracy found the best answer 33% of the time. Of the 33% of runs which it found the best answer the answer was on average found around iteration 6. The mean accuracy of the best Governor was 99.48%. The configuration of the Epistocracy algorithm was as follows:

- Council rate: 10%
- Crossover Rate: 50%
- Mutation Rate: 20%
- Tournament size: 5
- Population size: 20

To evaluate the performance and robustness of the Epistocracy algorithm, we compared our proposed algorithm with two state-of-the-art algorithms: Particle Swarm Optimization and Genetic Algorithm. These algorithms were also tasked to find the best model's hyper-parameters similar to Epistocracy. The population, iterations, and number of runs the algorithm tested at are given below:

- Population size: 20
- Iterations: 100
- Runs: 100

Table 4. Comparison of GA, PSO, and epistocracy algorithms using MNIST dataset.

	GA	PSO	Epistocracy
Mean accuracy of top best performing Governor	0.9948	0.9948	0.9948
Percentage of times the best answer was found	26%	28%	33%
Average number of iterations before finding the best answer	22.81	6.40	5.64

Particle Swarm Optimization. To compare how our algorithm performs against other algorithms, we used Particle Swarm Optimization (see Table 4). The configuration of PSO was the same as Epistocracy. The PSO mean accuracy was 99.48% and the best accuracy was found around the 7th iteration. The best accuracy was found 28 times out of 100 runs.

Genetic Algorithm. The other evolutionary algorithm tested was a genetic algorithm (see Table 4). The algorithm is a standard genetic algorithm using crossover and mutation.

The implementation found the best answer 26 times out of 100 runs. The mean accuracy of the best Individual was 99.48% and the best accuracy was found around the 23rd iteration.

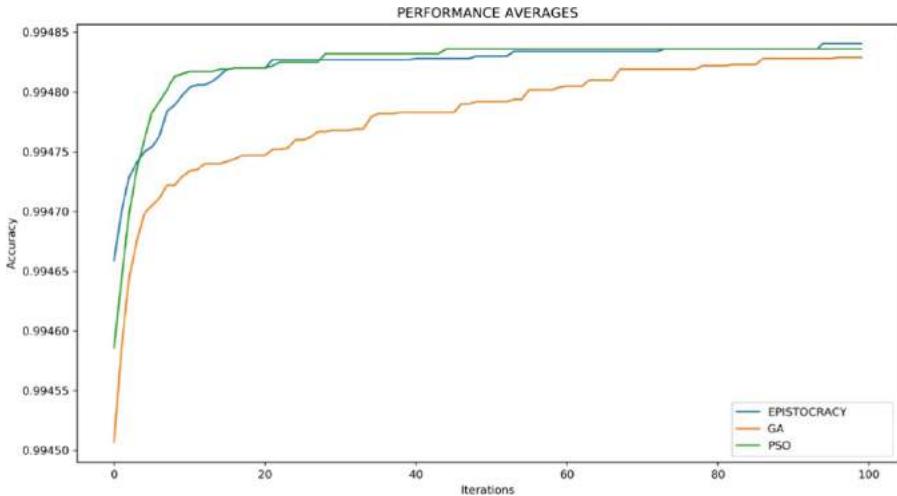


Fig. 4. Performance comparison of GA, PSO, and epistocracy.

Figure 4 shows that the Epistocracy algorithm initially has a higher accuracy than PSO and GA. After around 20 iterations, the Epistocracy algorithm asymptotically converges to the same fitness score of PSO, and eventually after about 92 iterations it defeats the PSO and shows higher accuracy. In this figure, the Epistocracy algorithm converges to the optimal solution faster than GA. However, even though PSO has a faster convergence rate of accuracy, it eventually fell behind Epistocracy. Overall, the Epistocracy algorithm shows better performance than the other algorithms.

4 Conclusion and Future Work

Evolutionary algorithms, in general, suffer from different types of problems such as premature convergence and stagnation which is closely related to the diversity of the population, curse of dimensionality and scalability, and a random, limited searching ability which usually occur in the absence of a guided change and due to unbalanced exploration-exploitation capacities.

This paper proposes a new multi-population evolutionary algorithm called Epistocracy based on socio-political evolution. In Epistocracy, there are two classes of population: governors and citizens. Citizens liberally follow governors to improve their performance through the exploration and exploitation of the search space. Governors, on the other hand, attempt to lead their population effectively to help the algorithm converge to the optimal solution in the early stages. Governors can be promoted or demoted

based on their population performance and votes. Individuals with better performance have votes of greater weights.

The Epistocracy algorithm was tested using several benchmark functions. The experimental results show that the Epistocracy algorithm can achieve superior results compared to other evolutionary and swarm-intelligence algorithms. Our proposed method is less likely to be trapped in local optima compared to other methods such as GA, PSO, ES, ABC, and CSA, and in some cases, can reach the optimal solution faster than existing algorithms. The Epistocracy algorithm uses the idea of rebels, dynamic resource management, gravitational force, and population variance to conduct an efficient explorative and exploitative search.

For future works, a number of research directions can be envisioned. First, the exploration-exploitation strategies can be enhanced to achieve a better convergence rate. Second, a multi-objective version of the algorithm can be implemented. Third, a more comprehensive test set with high dimensionality can be utilized, and the results compared with more evolutionary and swarm intelligence algorithms. Finally, the Epistocracy algorithm can be adapted for the discovery of optimal architectures of Convolutional Neural Networks and their hyper-parameters.

References

1. Ong, F., Milanfar, P., Getreuer, P.: Local kernels that approximate Bayesian regularization and proximal operators. *IEEE Trans. Image Process.* **28**(6), 3007–3019 (2019)
2. Holland, J.H.: *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, pp. viii, 183 p. University of Michigan Press, Ann Arbor (1975)
3. Holland, J.H.: Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.* **2**(2), 88–105 (1973)
4. Holland, J.H.: Outline for a logical theory of adaptive systems. *J. ACM (JACM)* **9**(3), 297–314 (1962)
5. Rechenberg, I.: Cybernetic solution path of an experimental problem: Kybernetische Lösungsansteuerung Einer Experiméntellen Forschungsaufgabe. RAE (1965)
6. Beyer, H.-G., Schwefel, H.-P.: Evolution strategies – a comprehensive introduction. *Nat. Comput.* **1**(1), 3–52 (2002). <https://doi.org/10.1023/A:1015059928466>
7. Schwefel, H. P.: Kybernetische evolution als strategie der experimentellen forschung in der strömungstechnik. Dipl.-Ing. Thesis (1965)
8. Eberhart, R., Kennedy, J.: Particle swarm optimization. In: *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948. Citeseer (1995)
9. Pan, X., Xue, L., Lu, Y., Sun, N.: Hybrid particle swarm optimization with simulated annealing. *Multimed. Tools Appl.* **78**(21), 29921–29936 (2018). <https://doi.org/10.1007/s11042-018-6602-4>
10. Wang, S., Liu, G., Gao, M., Cao, S., Guo, A., Wang, J.: Heterogeneous comprehensive learning and dynamic multi-swarm particle swarm optimizer with two mutation operators. *Inf. Sci.* **540**, 175–201 (2020). <https://doi.org/10.1016/j.ins.2020.06.027>
11. Ratnaweera, A., Halgamuge, S.K., Watson, H.C.: Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. Evol. Comput.* **8**(3), 240–255 (2004)

12. Suganthan, P.N.: Particle swarm optimiser with neighbourhood operator. In: Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), vol. 3, pp. 1958–1962. IEEE (1999)
13. Kumar, S., Nayyar, A., Kumari, R.: Arrhenius artificial bee colony algorithm. In: Bhattacharyya, S., Hassanien, A., Gupta, D., Khanna, A., Pan, I. (eds.) Innovative Computing and Communications, vol. 56, pp. 187–195. Springer, Heidelberg (2019). https://doi.org/10.1007/978-981-13-2354-6_21
14. Karaboga, D.: An idea based on honeybee swarm for numerical optimization. Technical report-tr06, Erciyes University, Engineering Faculty, Computer (2005)
15. Bajer, D., Zorić, B.: An effective refined artificial bee colony algorithm for numerical optimisation. Inf. Sci. **504**, 221–275 (2019). <https://doi.org/10.1016/j.ins.2019.07.022>
16. Xiao, S., Wang, W., Wang, H., Zhou, X.: A new artificial bee colony based on multiple search strategies and dimension selection. IEEE Access **7**, 133982–133995 (2019). <https://doi.org/10.1109/ACCESS.2019.2941247>
17. Yang, X.-S., Deb, S.: Cuckoo search via Lévy flights. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 210–214. IEEE (2009)
18. Mareli, M., Twala, B.: An adaptive Cuckoo search algorithm for optimisation. Appl. Comput. Inf. **14**(2), 107–115 (2018). <https://doi.org/10.1016/j.aci.2017.09.001>
19. Li, J., Li, Y.-X., Tian, S.-S., Xia, J.-L.: An improved cuckoo search algorithm with self-adaptive knowledge learning. Neural Comput. Appl. **32**(16), 11967–11997 (2019). <https://doi.org/10.1007/s00521-019-04178-w>
20. Li, H., Yuan, D., Ma, X., Cui, D., Cao, L.: Genetic algorithm for the optimization of features and neural networks in ECG signals classification. Sci. Rep. **7**, 41011 (2017)
21. Xi, W., Li, G., Moayedi, H., Nguyen, H.: A particle-based optimization of artificial neural network for earthquake-induced landslide assessment in Ludian county, China. Geomat. Nat. Haz. Risk **10**(1), 1750–1771 (2019)
22. Kim, T.S., Na, J.C., Kim, K.J.: Optimization of an autonomous car controller using a self-adaptive evolutionary strategy. Int. J. Adv. Rob. Syst. **9**(3), 73 (2012)
23. Prakash, M., Saranya, R., Jothi, K.R., Vigneshwaran, A.: An optimal job scheduling in grid using cuckoo algorithm. Int. J. Comput. Sci. Telecommun. **3**(2), 65–69 (2012)
24. Yeh, W.-C., Su, J.C., Hsieh, T.-J., Chih, M., Liu, S.-L.: Approximate reliability function based on wavelet latin hypercube sampling and bee recurrent neural network. IEEE Trans. Reliab. **60**(2), 404–414 (2011)
25. Contreras, R.C., Morandin Junior, O., Viana, M.S.: A new local search adaptive genetic algorithm for the pseudo-coloring problem. In: Tan, Y., Shi, Y., Tuba, M. (eds.) ICSI 2020. LNCS, vol. 12145, pp. 349–361. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53956-6_31
26. Rahmani, S., Amjady, N.: Non-deterministic optimal power flow considering the uncertainties of wind power and load demand by multi-objective information gap decision theory and directed search domain method. IET Renew. Power Gener. **12**(12), 1354–1365 (2018)
27. Han, M., Liu, C., Xing, J.: An evolutionary membrane algorithm for global numerical optimization problems. Inf. Sci. **276**, 219–241 (2014)
28. Kheshti, M., Kang, X., Li, J., Regulski, P., Terzija, V.: Lightning flash algorithm for solving non-convex combined emission economic dispatch with generator constraints. IET Gener. Transm. Distrib. **12**(1), 104–116 (2017)
29. Zhang, K., Li, B.: Cooperative coevolution with global search for large scale global optimization. In: 2012 IEEE Congress on Evolutionary Computation, pp. 1–7. IEEE (2012)
30. Vanaret, C., Gotteland, J.-B., Durand, N., Alliot, J.-M.: Preventing premature convergence and proving the optimality in evolutionary algorithms. In: Legrand, P., Corsini, M.M., Hao, J.K., Monmarché, N., Lutton, E., Schoenauer, M. (eds.) Artificial Evolution (Evolution Artificielle).

- LNCS, vol. 8752, pp. 29–40. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-319-11683-9_3
31. Dolson, E.L., Vostinar, A.E., Wiser, M.J., Ofria, C.: The MODES toolbox: measurements of open-ended dynamics in evolving systems. *Artif. Life* **25**(1), 50–73 (2019)
 32. Rajan, A., Malakar, T.: Optimal reactive power dispatch using hybrid Nelder-Mead simplex based firefly algorithm. *Int. J. Electr. Power Energy Syst.* **66**, 9–24 (2015)
 33. Sreejith, S., Nehemiah, H.K., Kannan, A.: Clinical data classification using an enhanced SMOTE and chaotic evolutionary feature selection. *Comput. Biol. Med.* **126**, 103991 (2020)
 34. Bardeen, M.: Survey of methods to prevent premature convergence in evolutionary algorithms. In: Workshop of Natural Computing, J. Chilenas de Computation, pp. 13–15 (2013)
 35. Jansen, T.: Analyzing Evolutionary Algorithms: The Computer Science Perspective. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-3-642-17339-4>
 36. Tahir, M., Tubaishat, A., Al-Obeidat, F., Shah, B., Halim, Z., Waqas, M.: A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare. *Neural Comput. Appl.* 1–22 (2020). <https://doi.org/10.1007/s00521-020-05347-y>
 37. Jadon, S.S., Tiwari, R., Sharma, H., Bansal, J.C.: Hybrid artificial bee colony algorithm with differential evolution. *Appl. Soft Comput.* **58**, 11–24 (2017)
 38. Murugan, R., Mohan, M., Rajan, C.C.A., Sundari, P.D., Arunachalam, S.: Hybridizing bat algorithm with artificial bee colony for combined heat and power economic dispatch. *Appl. Soft Comput.* **72**, 189–217 (2018)
 39. Kramer, O.: Premature convergence in constrained continuous search spaces. In: Rudolph, G., Jansen, T., Beume, N., Lucas, S., Poloni, C. (eds.) Parallel Problem Solving from Nature. PPSN 2008. LNCS, vol. 5199, pp. 62–71. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87700-4_7
 40. Mill, J.S.: Mill on Bentham and Coleridge. Chatto and Windus, London (1950). Leavis, F.R. (ed.)
 41. Brennan, J.: Against Democracy: New Preface. Princeton University Press (2017)
 42. McKay, M.D., Beckman, R.J., Conover, W.J.: Comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21**(2), 239–245 (1979)



An Approach for Non-deterministic and Automatic Detection of Learning Styles with Deep Belief Net

Maxwell Ndogkon Manga^{1(✉)} and Marcel Fouda Ndjodo²

¹ Department of Computer Science, Faculty of Sciences, University of Yaoundé I, P.O Box 812, Yaoundé, Cameroonian
manga.maxwell@univ-yaounde1.cm

² Department of Computer Science and Educative Technology, Advanced Teachers Training College, University of Yaoundé I, P.O Box 47, Yaoundé, Cameroonian

Abstract. In spite of the remarkable development of Learning Management Systems, the observation state that these systems don't fit student learning preferences. To reduce the dropout of student in online learning platforms, learning should be personalized. The migration from classical learning to personalized learning is one way to increase learners' abilities and improve their learning skills or preferences. This paper defines an approach for developing models of automatic and non-deterministic learning style detection, based on the traces of learner activity in adaptive learning systems. The approach interest lies, on one hand, in its automatic character of the detection of learning traces based on the interactions of learners' activity in the system instead of a questionnaire used to initialize student Learning styles. On the other hand, in its non-deterministic nature, not like deterministic approaches that do not take into account the uncertainty associated with learning traces, it also uses literature based approached to have an approximation of computed learning styles. Detection is automatic in that unlike traditional methods, it does not question learners to collect information about their learning styles. It is non-deterministic in that it considers the stochastic nature of the learning traces. In the developed approach, DBN-LIS, a generative algorithm, is used to analyse learning traces on LMS, in order to eliminate their inherent stochastic character. The results of this approach are effective, using, unlabelled Moodle and an expert's labelled dataset, were we reached 91.21% of right detections.

Keywords: Adaptive intelligent learning systems · Personalized learning · Learning style · Deep Belief Net · Feature selection

1 Introduction

1.1 Context

Learning Management Systems (LMS) are now oriented, and compliant to student learning experience. Adaptive and Intelligent Educational Systems (AIES) refers to technologies which place student at the centre of the learning process. It attempts to be adaptive

by building a model of student goals, preference and knowledge through a set of interactions with the systems to fit the student need [8, 31]. The study on adaptive e-learning systems emphasis personalized systems, such system provides a different view or actions in e-learning platforms basis on student preferences [9]. The set of such student cognitive and psychological characteristics involved during learning are gathered in a hierarchical concept which defines the learning style dimension of the student. Learning styles are set of cognitive, emotional characteristics and physiological factors that serve as the relatively stable indicators of how student perceives, interacts and respond to the learning system. It is one of the most important key factors for a successful e-learning experience. It permits to limit the dropout of student on online learning systems [2]. This dropout is mainly due by a lack of motivation on the learning materials or activities provide by the system and the monotonic learning process for all. In this sense, learning should be appealing and tailored to student preference in order to reduce their dropout. Learning style is a decision support for the teacher, to guides its pedagogy and ensure the right way of student learning process. For the students, it enriches the learning experience for the student through deliverance of shaped learning activities, moreover it reduces cognitive overload, through minimization of learning time [9, 24]. While tremendous benefits are associated with learning styles, their efficient computation and implementation can be far from supposed objectives and subject to some trends. Artificial intelligence models play an important role in many scientific fields they are found too in education [26], allowing new learning systems to cope with a learner interest and need. Learner centric education is one of the trends of the future e-learning system [18]. The transition from industrialized to personalized education would on one hand, enable a learner to benefit from rich learning activities and limit the duration of learning, on the other hand to increase his performance [9] through the construction of machine learning model for improving teaching and learning.

1.2 Problem

Learning styles initialization is a difficult process since even students don't know their learning strategy. Some collaborative tools as ILS (Index of Learning Styles) are used under questionnaire form to pull out their learning preferences. But it doesn't fit due to student auto conceptions or self-representations of their learning [7, 25]. This lead then in subjectivity when initializing learning styles and causes poor accuracy measurement and difficulty to update.

Learning styles are stochastics many studies [9, 20, 25] argued that learning styles could vary over time and be dependent on learning materials. This nonstationary aspect is due to learning traces or patterns which varies considerably regarding learning content format. So, learning traces registered on e-learning system contains inner drawback due to their non-deterministic nature. By this way the traces let on systems should not been taken as certainties.

Data-driven approaches are used to infer learning styles in e-learning systems, Resent years. They grant some reliability in the learning style detection process. Computer intelligent algorithms are used for the dynamic student modeling strategy. Amongst them we have deterministic algorithms, which used expert to build student learning style. Experts build learning questionnaire to have information about student learning compliance and

the learning environment. Such experts used generally questionnaire. Non-deterministic algorithms are less dependent on expert and generally based on student experience on the system. And tend to build autonomously the student learning model preference.

This hybrid approach will use theoretical learning style definition, the Felder and Silverman Learning Style Model (FSLSM), which is the must use on regular studies involving learning styles on e-learning platforms [15]. The ILS questionnaire will be refined by new computer intelligence algorithms to ensure reliability and authoring of learning style inference.

The related model DBN-LIS (Deep Belief Net - Less Initialization Student) uses Deep Belief Network a class of generative algorithms in deep learning to improve the generalization capacity of the detection models built. One of the advantages of such generative networks is that they are nondeterministic by nature and will try to build targeted learning styles autonomously in an unsupervised manner without the presence of an expert domain. Just by learning distribution on learning traces and having the ability to understand the distribution law of the data.

1.3 Objectives

The main contributions of this paper are as follow:

1. An efficient non-deterministic and automatic detection of learning styles model, to improve learning on adaptive e-learning systems. This model will analyse the behaviour of a learner, in order to dynamically detect his learning style and consequently, adapts its training activities. It will be independent on questionnaire learning style initialization.
2. A generic architecture above Learning Management Systems (LMS) to ensure the efficiency personalization of learning activities.

The remainder of this paper is structured as follows. In Sect. 2, we offer a succinct overview of the literature and theory review. For Sect. 3, we present the theoretical framework. In Sect. 4, the description of the research methodology and the procedure of this research are presented. Section 5 provides findings on the research objectives. We conclude the work with discussions, limitations, and future study suggestions.

2 Literature and Theory Review

In this section we will first illustrate the concept of learning style and the relevant theoretical model commonly used to specify student model. We then list several approaches used to infer learning styles.

2.1 Understanding Learning Styles

A learning style describes the behaviour adopted by a learner during a learning activity [7, 21]. This is the way in which he analyses and understands the concepts of a learning activity. It has three dimensions:

1. Perceptive, the way in which he perceives the information presented to him,
2. Cognitive the set of actions that the learner can mobilize to process the information
3. Metacognitive, referring to the ability to reason about perceived information.

It can vary from learner to learner, as well as not all learners react in the same way to the information presented to them. For example, some learners effectively conceptualize when the information they perceive is represented by illustrations, others when confronted with exercises, some learners require step by step learning: iterative and sequential. A learning style itself can thus be seen as, an abstraction of a learner's learning process. There are several theoretical models of learning style: Felder and Silverman (FSLSM, KOLB, VARK). Of all these models, the one that is used the most in e-learning is FSLSM [7, 15].

The authors in [32, 34] identify and propose a categorization of these learning styles. They prescribe that the individual characteristics of learners must be taken into account before any training activity, in order to ensure useful and effective learning. The work of Felder and Silverman is more generic and incorporates other learning style categorization approaches [25].

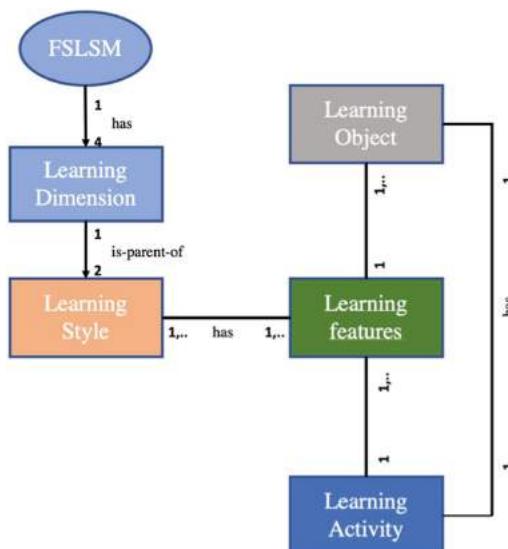


Fig. 1. Ontological view of the Felder and Silverman learning style model

The FLSM has four (04) learning dimensions, as shown in Fig. 1, each of dimension contains two (02) learning styles poles. Each learning style is associated with a set of patterns configurable on the different assets of LMS from the learning activities to the learning content. These features are variables which will catch student learning traces on the system and will permit then to have a representation of the student mode. Felder's learning dimensions are follows:

Processing: The processing of information by a learner is carried out exclusively in “**Active/Reflective**” ways. The qualifier “Active” characterizes a learner who regularly interacts with the learning materials and who regularly explores the information he receives. He also interacts with his peers; he is indeed an active learner. Unlike the qualifier “Reflective” characterizing a learner who conducts a deep reflection on the information received before any action.

Perception: The perception of information by a learner is done in a “**Sensing/Intuitive (S/I)**”. The “Sensing” category characterizes a learner who perceives information through meaning; he learns from the facts. He enjoys using standard problem solving or case study approaches. He is a pragmatic learner; he only understands information if it can be materialized. Unlike the qualifier “Intuitive”, which characterizes a learner with strong abstraction capacities, he has a developed intuition, and prefers abstract learning materials such as theories. He can manipulate concepts and represent them at different levels of knowledge. He is an intuitive learner or even a creative one.

Presentation: The presentation of information by a learner is viewed from “**Visual/Verbal (V/V)**” angles. The qualifier “Visual” designates a learner who learns easily when information is presented to him/her in the form of figures, illustrations or diagrams; it is a learner with visual memory. Unlike a “Verbal” type learner who is a learner who learns best by listening to the tutor of the course and the explanations.

Understanding: Understanding Information describes the “**Sequential/Global (S/G)**” sub-dimensions. The qualifier “Sequential” characterizes a learner who learns step by step, in a linear manner, information must be offered stepwise. The “Global” dimension characterizes a learner who can randomly learn several concepts at the same time in a non-linear fashion.

2.2 Theory and Past Research

The review [15] analyses the research activity on the detection of learning styles over the last 20 years, several approaches have been used, moving from classical to dynamic approaches. Classical approaches do not require the modeling of learning profiles, they are based on collaborative techniques to simple rule systems [21, 25]. Dynamic approaches [5, 7, 15, 28] take into account all the interaction mechanisms in the e-learning systems: consultation of pages, answers to questionnaires, discussion, etc. All of these mechanisms constitute representations of a learner’s learning within an online training system; These representations vary from one learner to another depending on the learning styles involved, and subject to study. The dynamic approaches are data-driven, they use student behaviour data, represent as traces in e-learning platforms: Amongst these dynamic approaches, we can also mention [7] who used an augmented decision tree to infer learning styles, in other way, [9, 30] used Multi-Layer Perceptron to infer learning preference of student. Novel approaches [5, 7, 9] are hybrid relying classical approaches, referring for example to Index of Learning Styles (ILS) and dynamic approaches, this strategy introduce improve results but have some limitation regarding the stochastic nature of learning styles [12, 20, 25].

New driven data approaches are more efficient in the detection of FSLSM. Some take advantages of the data size provided by some LMS, others by taking the advantages of new trends of computer intelligence algorithms we can illustrate them as follows:

In [1], the author designed a deep multi target using Artificial Neural Network (ANN) to infer learning styles, the system starts by feeding the Index of Learning Styles questionnaire (ILS). Then model student learning descriptors students input to infer multiple Learning styles. Instead of detecting one learning style of the FSLSM, it infers all of them at the time. The results shown that, ANN with multiple targets improve better comparatively to the work of [9].

In [3], the author built a Deep Belief Network (DBN) to detect learning styles. The student model features, learning indicators and targets were given using the experts experience. They setup the ILS before passing the data to the DBN model to learn on data. This DBN model reach a performance of 87.7% of accuracy. Comparatively to backpropagation ANN, DBN improve results quality with less standard deviation on accuracy percentage.

In [4], the author conceived a Fuzzy C Mean (FCM) to infer Felder and Silverman, an important of this paper is that instead of detecting learning style only using one course, it uses many courses to classify learning styles, this approach made identification more robust. The performance reached by FCM is improved 93.41% of detection.

A decision tree model to infer three (03) learning dimensions of the FSLSM was proposed in [2]. Results achieved was 98%, It uses unsupervised strategy by clustering the data and inferring the target of those cluster before applying hyper optimization on Decision Tree (DT) parameters. It takes advantages of data collected from the EdX courseware where more than 52000 students where registered.

In order to personalize learning in e-learning, several approaches have been illustrated, particularly recommendation systems [22], but these are cost effective because they force the type of interaction that the learner can have on the platform, and they associate other persons to recommend kind of learning objects used by learner. Thus, limiting the centric and oriented nature of the student learning process.

From the given review one of the limits we found, is the initialization of learning styles by using Index of Learning Style (ILS), which lead us to two issues: In one part, the cost to initialize or labelling student learning styles, it requires an expert to annotate student learning preferences and in other part, the high potential of subjectivity in students' answers to the ILS questionnaire [20].

Learning styles have a stochastic nature within them [25]. The fact that supervised approaches recommend the use of prototype ILS questionnaire to initialize learning styles to adjust the learning phase of models inadvertently induces some error as uncertainty. The learner who completes a form in platform user's interface can answer questions without being objective in its answers or having subjective answers [12, 20]; this decreases the certainty of estimating or approximating the learner's actual learning style and therefore limits his learning on the online training platform.

Also noteworthy is the fact that there are learners who do not have a dominant learning style and who conform to any form of learning strategy. These properties make complex the automatic and efficient detection of learning styles. Optimizing the detection of learning styles should therefore consider the following three points: the non-determinism

of learning styles (stochastic character), the initialization cost of the learning styles and the absence of unanimous learning features and size required to model them.

The definition of a non-supervised modeling adaptive e-learning system using student's learning traces would address this limit.

3 Theoretical Framework and Hypothesis

3.1 Automatic Learning Styles Estimation

The starting point of modeling learning styles is their prior estimation. Researchers stand out with ILS to have initial knowledge of student learning style [1, 3, 8, 9] this strategy has a cost in time and resources as it requires sufficient delay to permits student to respond to the questionnaire and pre-process and compute learning styles of the students, it also involves experts to setup learning indicators and questions for student interview [3]. We posit the succeeding assumption to enhance the cost of learning styles initialization this will rely from students' inputs in the learning systems a prior of his learning preferences.

Hypothesis 1. The automatic computation of data target reduces the initialization cost of learning styles.

Hypothesis 2. The automatic initialization avoids subjectivity and autoconceptions on student's learning preference.

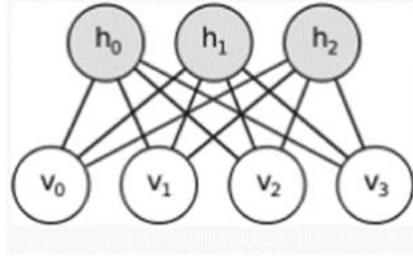
3.2 Nondeterministic Learning Traces

In real life, data doesn't traduce often the reality. It may be possible that they are uncertain or variant this intuition supports the aim of [20, 25] who state that learning styles are stochastic. Therefore, we can expect that some approaches as supervised learning, which learn from labels, could failed during the inference of learning styles. The problem is elsewhere, it is non-deterministic and the approaches which deal well with this kind of problem in artificial intelligence are unsupervised.

3.3 Deep Belief Nets

It belongs to the larger set of deep neural networks; it is a generative graphical model using probabilities to learn on data. A DBN can limit the problems associated with the exploding weights of the gradient method observed in the highly connected Multi-Layer Perceptron (MLP) learning algorithm [6, 17, 29], which is used in learning styles detection by [1, 5, 9, 30]. DBN can make possible to reconstruct the input data, from the modeling distribution of the latent variables. It is composed of several structured layers called Restricted Boltzmann Machines (RBM).

An RBM is a stochastic neural network, capable of being trained in unsupervised mode to generate samples according to a distribution of complex probability specified by examples [29]. It has a visible layer (v neuron units) and a hidden layer, consisting of hidden or latent variables [6] these layers are interconnected by weights.

**Fig. 2.** A RBM graph model

The RBM structure in Fig. 2 will construct a representation of learning traces data v , by encoding the variables of v to obtain h which will be an abstract representation of v resulting from the learning of the distribution of learning data. This learning is defined by minimizing an energy function $E(v, h)$ of each edge in the graph model,

$$E(v, h) = \sum_i a_i v_i + \sum_k b_k h_k + \sum_i \sum_j w_{i,j} v_j \quad (1)$$

With $v_i, h_i \in \{0, 1\}$

The parameters of the model to be discovered by the algorithm being $\theta = < a, b, w >$. The input variables of the layer v of an RBM are independent of each other, hence the absence of a link between it. The encoding of the layer from v is defined by the joint probability distribution of v on a neuron of h by:

$$p(v, h) = \frac{e^{-E(v, h)}}{Z} \quad (2)$$

The Z function is a partition function that normalizes this probability. The marginal distribution on layer h , from the inputs.

$$p_\theta(v) = \sum_h p(v, h) \quad (3)$$

The objective of the optimization of θ is to maximize the resemblance between the marginal distribution $p_\theta(v)$ generated by the RBM and the experimental distribution observed within the training set E . For this reason, we will maximize the likelihood

$$Vr(x) = \int_{x \in E} p_\theta(x) \quad (4)$$

Or by using its corollary, the optimization of the log likelihood

$$IVr(x) = - \sum_{x \in E} \log p_\theta(x) \quad (5)$$

Reconstruction cost of the visible units of an RBM using binary cross entropy is:

$$C \left\{ x, \tilde{x} \right\} = - \sum_i x_i \log \left\{ \tilde{x}_i \right\} + (1 - x_i) \log \left\{ 1 - \tilde{x}_i \right\} \quad (6)$$

RBM Learning. Unlike other neural network algorithms based on gradient backpropagation, a DBN has a specific learning algorithm. [29] proposes the contrast divergence (CD) which is a probabilistic algorithm that will generate sequences of samples approximating with efficiency hidden layers unit in h according to the marginal probability $p\theta(v)$ observed in the training data. It makes it possible to maximize the likelihood in Eq. 4, to do so, it would be necessary to define the gradient of $L_{Vr}(x)$ to optimize the parameters θ of the model.

3.4 Conceptual Framework

From review we have observed that the general design of adaptive e-learning System using learning styles has many components. In Fig. 3, we show the purposed the new architecture based on data-driven and non-deterministic learning style. The Instructional environment provides to student the viability to learn from remote and in pace way from their computers or devices; it can be Blended Learning Environment or Distance Learning. It contains the Learning Management System Frontend (LMS-frontend) which present web pages to the user, and the LMS system Backend which contains the database and all the business logic of the platform. The instructional model which organizes the learning activities and learning objects. It contains, the Instructional design module. The student model which is characterized by the student profile and its interactions between the instructional environment, and the instructional model.

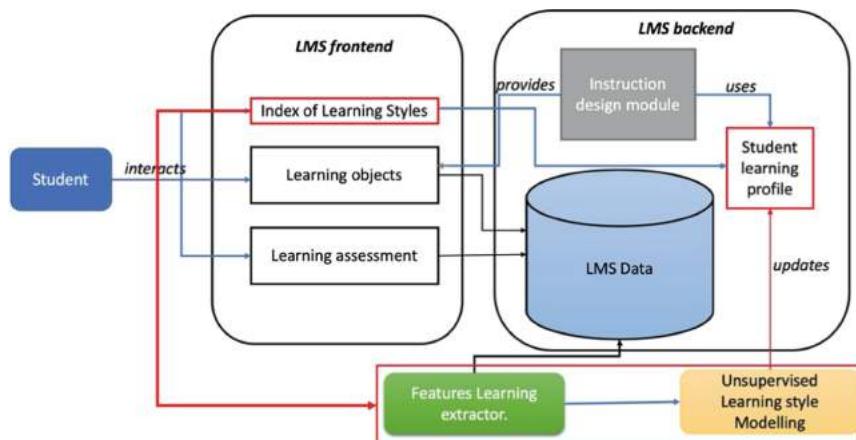


Fig. 3. Architecture of adaptive e-learning system based on learning style

The connected learner is subjected to a first phase of the ILS [32] which allows to have a system initialization for each learner. Once done, the learner can explore and interact with the educational content (Learning object, activities, assessment,) available in several formats and offered by the component Instruction design module, it can be for example a SCORM component. The learner's interactions stored in the database are then extracted. A pre-treatment phase is then performed to prepare the learning indicators to be

modelled. The model will analyse the distribution of learning indicators and dynamically infer the learning style. Once detected, the model will update the learning profile of this learner, it will have two roles in the platform modeling the traces of learning, as well as learning profiles. The design training module will refer to the learner's learning profile and provide him with learning activities appropriate to his learning style.

For this research amongst all the components in Fig. 3, we will prune ILS as shown in Fig. 4, and focus on the student learning traces and the unsupervised model approach. Previous studies used ILS as an expert knowledge to classify student learning style but in real word when practicing it is difficult to them to be canonical because they contain some uncertainty so the proposed architecture of adaptive LMS is given by:

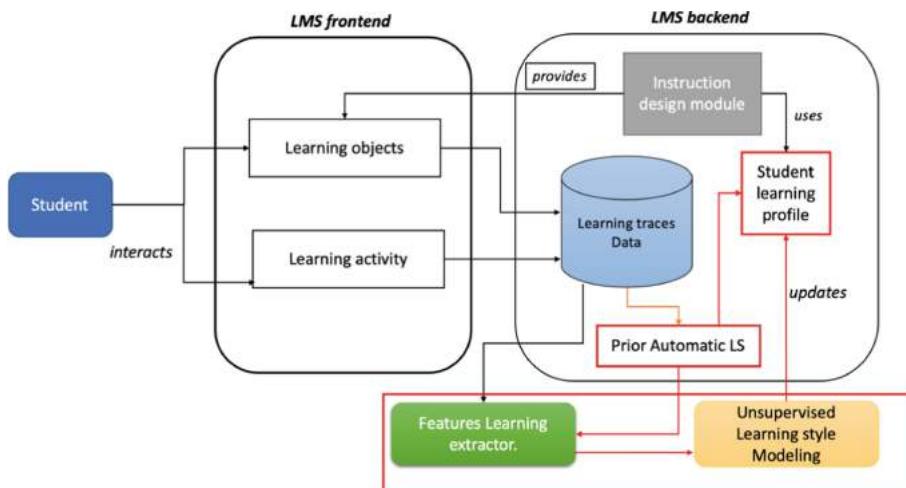


Fig. 4. Proposed architecture for automatic and unsupervised learning styles detection

We use this prior estimation of Learning style to initialize the student model: learning descriptors in database and student profile, to build another form of ILS but nondeterministic as the learning will be considered as unsupervised and stochastic. By just focusing on learner interactions and on real time.

4 Research Methodology

The methodology we defined in Fig. 5 goes from data gathering and pre-processing to model evaluation. The related method is described as follows:

The first stage is to extract the data from the learning system, then a pre-processing stage is followed to build student learning descriptors preferences matrix using a clustering algorithm, this algorithm permits us to discover automatic thresholds for each learning trace values in its corresponding learning dimension. In previous studies [9, 20, 25] used experimental thresholds to discover preference or hint student.

The second stage is where the prior student learning styles of students are built. This automatic strategy uses the Deles [25] approach to build an estimation of labelled

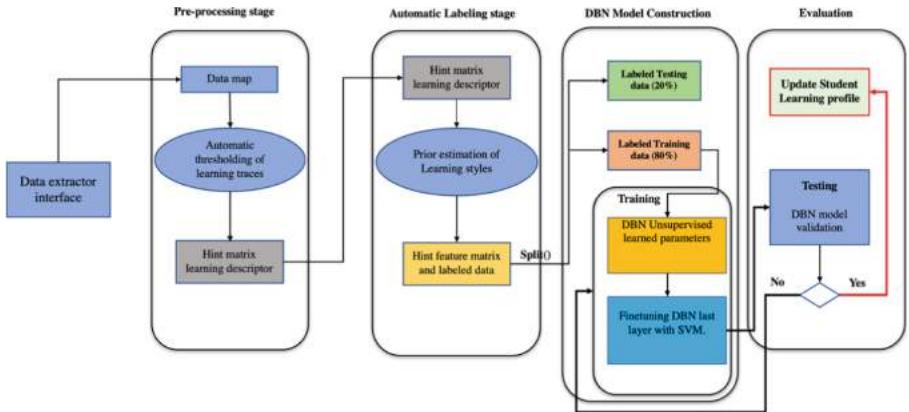


Fig. 5. Proposed detection method for automatic initialization and detection of learning styles using deep belief networks.

data. Using this technique, it will be no more mandatory to use ILS questionnaire, one of this advantage is that we compute apriori student learning styles based on their real experience with the system instead of having their answer on a form which are potentially subjective.

The Third stage is the unsupervised modeling by nature DBN deals with stochastic variable and it is a good point to model learning traces with this algorithm which will learn first from data to generate the distribution of learning indicators then in second phase will used supervised algorithm for transferring the knowledge of learnt parameters.

The last stage will be the model evaluation, which will decide if we can use the model for the adaption of student profile. If The performance of the model is good, it can be used to reset student learning style in its student profile.

4.1 Data Collection

This research method, we used Felder & Silverman Learning Style Model (FSLSM) they are mainly used on e-learning learning styles as related in [25]. We noticed in general, a strong insufficiency of public tracking data on e-learning systems. that captures interactions between learners and the system; review data are for the most part of time proprietary then do not have public access, also. Having a large dataset is a benefit for deep learning model to improve generalization of data [6, 29]. To overcome this shortcoming, we used two datasets, the first dataset, without labelled learning styles, uses the traces of the Moodle learning system data for research, which is open and sufficiently provided to experiment a fully online learning format. 2168 students agreed to use their data for research purposes [10]. The second dataset uses labelled learning styles provide by the experts. The learning traces comes from Philippines college with 507 students [11], author provide too a decision tree algorithm to model Felder and Silverman learning styles in this dataset.

Learning Descriptor. On the analysis of the Moodle logfile, we have been able to identify the 4 dimensions of Felder and Silverman learning styles (FSLSM). We want

to notice that this method is generic and can be apply to other theoretical learning style model (VARK, FSLSM...). From the learning activities found in the database we have parted the Moodle modules present according to the learning style dimensions of Felder and Silverman, in Fig. 6 we can observe the learning traces extracted for the active and reflective learning style dimension.

Table 1. Moodle modules repartition through Felder and Silverman learning style model.

Active/Reflexive	Sensitive/Intuitive	Visual/Verbal	Sequential/Global
Forum	Feedback	Glossary	Page
Quiz	Book	Lesson	Wiki
Chat	Choice	Forum	Lesson
Lesson	Lesson	Bigbluebutton	Choice
Survey			
Workshop			

In this log file, we found 13 (thirteen) activity modules. These modules are distributed in the 3 Felder & Silverman Learning Style Model retrieved in the log file of this system. See Table 1, this distribution follows the work guidelines of [14, 25], which suggest to group learning modules according each learning style dimension. For all these modules, we extracted 124 journal events interactions to pre-process the learning traces of each learner, see Fig. 6. These interactions constitute the learning variables of the detection model.

A	B	C	D	E	F	G	H	I	J	K
course_views	module	vicussion	vieventcourse	riscussion	sipevents	subevents	assessments	eventcourse	entcourse	mplevents
9	2	4	4	3	5	4	8	5	2	3
103	6	64	52	21	16	26	14	5	3	6
58	124	40	68	120	12	120	78	20	28	42
23	2	14	9	2	5	2	8	5	2	3
6	2	9	3	3	5	4	8	5	2	5
25	2	4	6	3	5	4	8	5	2	3
14	2	20	19	1	5	1	8	5	2	3
27	2	3	5	1	5	1	8	5	2	3
42	2	7	11	1	4	3	4	5	2	3
9	2	10	1	3	5	4	8	5	2	3
7	2	3	3	1	5	2	8	5	2	3
122	1	5	9	1	5	1	8	5	4	3
46	3	10	13	3	5	6	2	3	2	3
62	2	31	29	5	1	5	10	9	2	3
60	2	20	21	10	1	10	8	2	3	3
122	11	62	50	15	24	9	15	43	5	7
16	2	2	6	3	5	4	8	5	2	3
25	1	37	30	2	5	2	1	5	7	3
85	2	14	17	3	1	8	6	15	2	3
80	2	5	12	1	10	2	4	5	4	3
4	2	10	2	3	5	4	8	5	3	3
9	2	1	2	3	5	4	8	5	2	3

Fig. 6. Extracted learning traces for the processing dimension.

Pre-processing. Pattern behaviour selection. The Pareto curve in Fig. 7 describes the distribution of decreasing cumulative frequency learning indicators. From this curve, we

used learning features having considerable traces frequency (higher than 80%), from the 63 learning features present in the log file of the database. We applied the Algorithm 1. to extract the learning indicators variables. Before passing the data to learning models it is necessary to identify the most relevant learning traces in the data see Table 2. To make it effective, we calculate the maximum learning indicator of each learner, and make the disjoint meeting of them; this ensure that each learner has at list one learning preference in the system, then we select features which are related to the learning style prior estimation by visualizing if there is a relationship between them. This characteristic vector defines the representative variables for training the model.

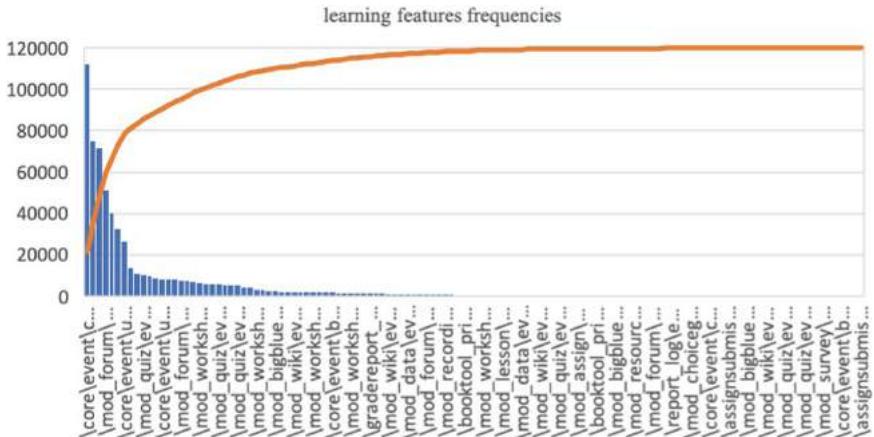


Fig. 7. Frequency of learning traces

For the Active/Reflective learning style, after crossing and analysing the high-input descriptors for the detection model, we identified 11 variables as models' learning inputs. For example, we see from Fig. 8(a).

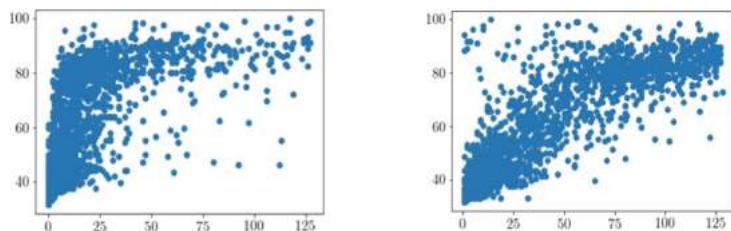


Fig. 8. (a) Left, forum learning traces (`mod_forum_event_discussion_viewed`) impact on processing dimension. (b) Right, chat traces Impact (`mod_chat_event_discussion_viewed`) on processing dimension.

That there is a correlation between the number of participation in the forums and the learning style category Active/Reflexive, it is the same observation in Fig. 8(b). For

the visits' number in the course content according to the learning style. This method was applied to the Felder & Silverman learning style dimensions to identify significant variables for the learning style detection models. From this analysis we also noticed that the learners are not categorized by the Perception dimension type because the learning traces of this dimension have a poor distribution.

Table 2. Moodle learning patterns after selecting relevant feature for each learning FSLSM.

Active/Reflective (A/R)	Visual/Verbal (V/V)	Sequential/Global S/G
-core_event_course_viewed	-mod_forum_event_course_module_viewed	-mod_page_event_course_module_viewed
-mod_folder_event_course_module_viewed	-mod_forum_event_discussion_viewed	-mod_wiki_event_page_viewed
-mod_forum_event_discussion_viewed	-mod_glossary_event_course_module_viewed	-mod_wiki_event_page_history_viewed
-mod_forum_event_course_module_viewed	-mod_forum_event_discussion_subscription_created	-mod_choice_event_course_module_viewed
-mod_forum_event_discussion_subscription_created	-mod_bigbluebuttonbn_event_bigbluebuttonbn_activity_viewed	-mod_choice_event_answer_submitted
-mod_workshop_event_submission_viewed	-mod_forum_event_course_searched	-mod_wiki_event_course_module_viewed
-mod_forum_event_assessable_uploaded	-mod_forum_event_assessable_uploaded	-mod_lesson_event_content_page_viewed
-mod_workshop_event_course_module_viewed	-mod_recordingsbn_event_recordingsbn_resource_page_viewed	
-mod_data_event_course_module_viewed	-mod_lesson_event_content_page_viewed	
-mod_choicegroup_event_course_module_viewed		
-mod_chat_event_sessions_viewed		

4.2 Proposed Technique of Data Analysis: DBN-LIS Method

To infer dynamically learning styles, we will need to extract the traces of learning activities of each learner in the system. Online learning systems can store these different activities in database. Once the extraction of the learning characteristics has been done, the analysis of the features describing learning in the system is possible.

Extraction of the Activity Traces on the Online Learning Platform. The learning system saves the entire learner's interactions; these traces define the behaviour of the learner on the platform. Each trace was categorized according to its belonging to a learning style dimension. This preliminary step consists of defining learning variables or indicators that are significant for learner monitoring on the platform.

We present a generic algorithm, not dependent of a learning platform, but interaction variables of learning styles present in a learning system, in order to extract the matrix of system learning indicators.

S The set of learners.

I The set of possible interactions in a learning system

LS The set of learning styles,

A include in **LS**, the set of features related to a learning style.

a_{ij} in **A**, is a learning traces, where the j-th learning features corresponds to the i-th learning style dimension.

hint(a_{ij}) Is a learning preference level [9, 25] this indicator illustrates the interaction's frequency of a learner on a_{ij}

$$\Gamma = \bigcup_{s \in S} X_s \quad (7)$$

The set of interactions of all the students enrolled to the course.

Algorithm 1: Learning style feature behaviour extraction

```

 $\Gamma = \{\}$ 
Foreach  $s \in S$ :
   $X_s = \{\}$ 
  Foreach  $a_{ij} \in A$  :
    if interact ( $s, a_{ij}$ )
       $X_s = X_s \cup \text{hint}(a_{ij})$ 
   $\Gamma = \Gamma \cup X_s$ 

```

For each learner in the system, we extract the interactions corresponding to the learning style dimension involved. From these traces, we calculate the associated learning indicators; these are defined by the function **hint**

$$\begin{aligned} \text{hint} : & LS \times A \rightarrow C \\ & (x, y) \mapsto h(x, y) \end{aligned} \quad (8)$$

C define a discrete space with value in {0, 1, 2, 3}, describing the level of preference of a learner in an interaction (learning activity). 3 means for a given learning style x in LS, the student has a *good* preference for the learning pattern y in A. To obtain those discrete group instead of manual thresholds defined by the experts, and used by [9, 20, 25] we use the clustering method k-means++ [27], to group student leaning traces by level of preferences and therefore build the partition C, defined by the below matrix.

$$\text{Hint} = \begin{matrix} & C_1 & C_2 & \dots & C_j & \dots & C_n \\ \begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_i \\ \vdots \\ S_m \end{matrix} & \left[\begin{matrix} h(a_{1,1}) & h(a_{1,2}) & \dots & h(a_{1,j}) & \dots & h(a_{1,n}) \\ h(a_{2,1}) & h(a_{2,2}) & \dots & h(a_{2,j}) & \dots & h(a_{2,n}) \\ \vdots & \vdots & & \vdots & & \vdots \\ h(a_{i,1}) & h(a_{i,2}) & \dots & h(a_{i,j}) & \dots & h(a_{i,n}) \\ \vdots & \vdots & & \vdots & & \vdots \\ h(a_{m,1}) & h(a_{m,2}) & \dots & h(a_{m,j}) & \dots & h(a_{m,n}) \end{matrix} \right] \end{matrix} \quad (9)$$

S_i design the i-th student.

$h(a_{i,j})$ is S_i level of preference on the related learning pattern A_i .

C_j is the set of student's level preferences on the learning pattern A_i . Each $h(ai,j)$ is a value or cluster label of C_j .

When the hint of student is defined, we can then have a prior consideration of the student learning style.

Literature Based Learning Styles Approximation. [13, 25] have shown that it is possible to have a prior estimate of a learner's learning styles using literate approach and using them for initialize the dynamic detection process. This calculation is done experimentally by weighting the resultant interactions of each learner.

$$LS_i = \frac{\sum_{j=1}^{P_i} hint_{i,j}}{n} \quad (10)$$

This computation can be used to estimate learning style using the mean and the standard deviation of LS_i distribution as shown in Table 3:

Table 3. Standard deviation vs Felder's scale values [23]

Possible value of student behaviour	Corresponded Felder's Value
μ	6
$\mu + 1\sigma$	7–8
$\mu + 2\sigma$	9–10
$\geq(\mu + 3\sigma)$	11
$\mu - 1\sigma$	5–4
$\mu - 2\sigma$	3–2
$\leq(\mu - 3\sigma)$	1

Equation 10 will down sample the Felder and Silverman Learning Style scale [-11, + 11[to restricted scale [0,1[and the related prior can be defined by the probability density function in Fig. 9:

To establish student learning style, we take into account the standard deviation and the mean of the LS approximation [23]. Student with high prior will be considered as positive behaviour for the related Felder and Silverman dimension's pole, and less prior will be considered as positive behaviour of the same dimension in the opposite pole, since those poles are symmetric.

$$label_LS_i = \begin{cases} 1 & \forall ls_i \in [\mu + \alpha\sigma, 1[\\ 0 & \forall ls_i \in [0, \mu - \alpha\sigma] \end{cases} \quad (11)$$

The scale:

- [$\mu + \alpha\sigma, 1[$] Represent students with strong positive learning style (e.g., Active)
- [$0, \mu - \alpha\sigma]$ Represent students with strong negative learning style (e.g., Reflective)
- [$\mu - \alpha\sigma; \mu + \alpha\sigma]$ Represent students with balanced learning styles.

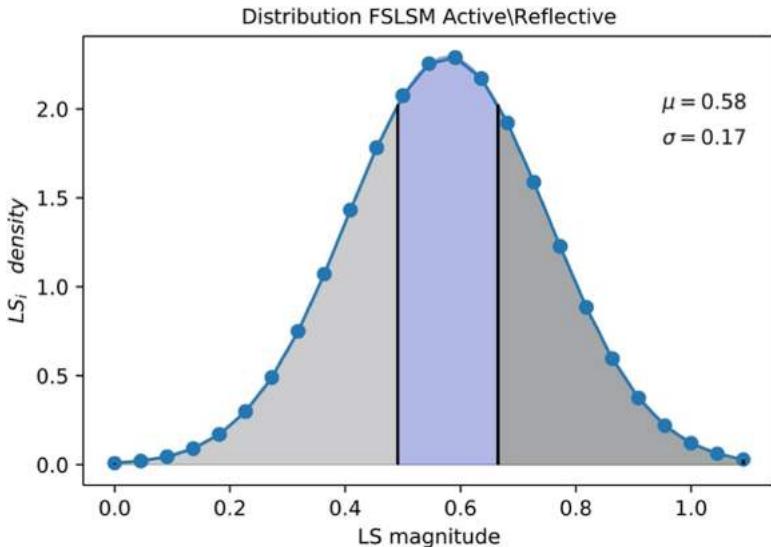


Fig. 9. Probability density of the processing dimension

Students with balanced learning style have density distribution in the neighbourhood of the mean according to the size of the standard deviation.

According to Fig. 9, we see that at the left side of student with balanced learning style, we have the negative pole of the learning styles in the processing dimension, this is the reflective pole. At the right of balanced student, we have the active pole. The student population at these two sides will be used to train model as their labels are estimated using Eq. 11. We remove students with balanced learning style in the modeling process. These students are less sensitive for any pedagogical strategy. They can adapt their learning independently of the learning features configure in the LMS.

As this distribution is empiric, to improve the inference we construct a deep belief nets which will learn or model this distribution, and then enhance the likelihood of this prior estimation.

4.3 Model Design

We have seen in Sect. 2.2 the different methods available for the automatic detection of learning styles and that, for an efficient computation, it is necessary to understand the learning behaviour of a learner on the platform without however definitively assigning a learning style questionnaire. We must use an unlabelled learning algorithm based on groupings. In order to group individuals in class of the same category and deduce learning styles, the latter emerging from themselves.

The unsupervised learning algorithm used in this article belongs to the family of deep neural networks: Deep Belief Network [6, 29]. It is a probabilistic algorithm that will analyse the probability distribution of the learning indicators constructed in Eq. 7; on the one hand to effectively estimate the notion of uncertainty related to learning style,

on the other hand to build a learning behaviour recognition's model of the learner. Our adapted DBN model includes an unsupervised learning layer consisting of RBM, and a supervised learning layer with a Support Vector Machine function to classify learning styles.

We define an isomorphic function $\text{discretize}()$ that takes hint indicators as input and transforms it into a d-bit sequence. It is important to discretize here to reduce some numerical computation issues as overflow/underflow when optimizing the parameters of the model, and this prepare the visible units to be used by the DBN model.

$$\begin{aligned} \text{discretize} : & A \rightarrow \{0, 1\}^d \\ (x) &\mapsto \text{discretize}(x) \end{aligned} \quad (12)$$

The new characteristic vectors' visible layer of DBN will be ($\text{discretize} \circ \text{hint}$) (x).

The algorithm generates a hybrid DBN model that learns the distribution of learning traces on a dataset. This DBN-based model can detect learner learning styles effectively by excluding bias due to uncertainty or stochasticity.

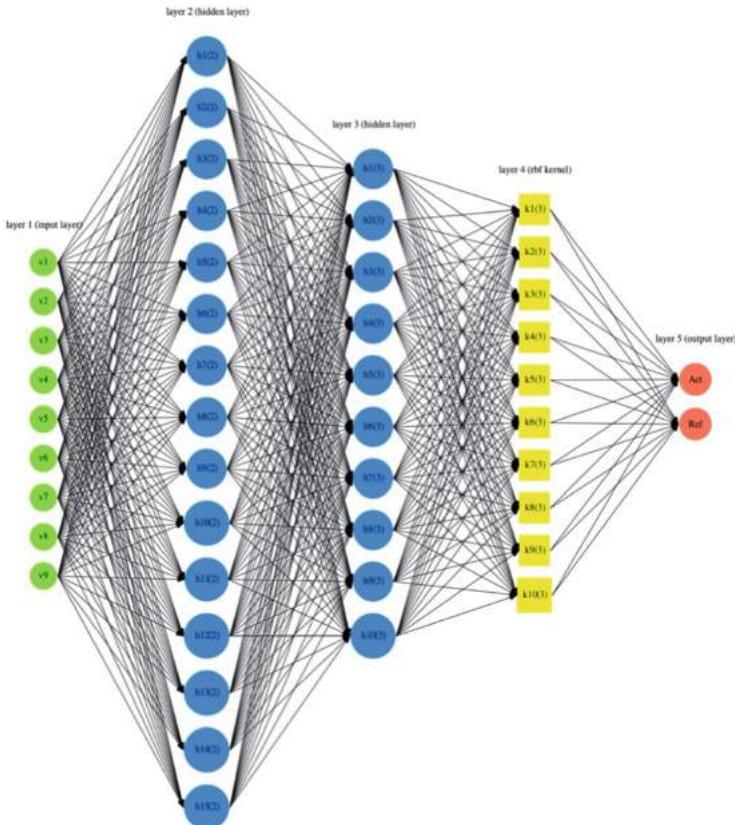


Fig. 10. A hybrid deep belief network to infer processing learning style

The first part of the model uses unsupervised learning, block 3 of the Algorithm 2, (see Annexe A.) to permit the model to learn the posterior density function distribution on learning style's features descriptor. At this level the contrastive divergence algorithm will be capable to analyse the pattern introduce in the model regarding either the uncertainty of visible units or their stochasticity, as a probabilistic model it will perform to learn latent transitions h_i of the variable v_i during the learning of parameters w, b, a ; using equations in Sect. 3.3, to estimate another pattern behaviour defining the generative input of the model, which can have a better stability according to the dataset distribution. And as dataset is bigger as the algorithm breakdown all this different transition to have the best similarity between the visible input and the generative one [6, 17] thus ensuring the confidence of the layer containing the generative input. The second part of the model is a Support Vector Machines (SVM) [33] with radius basis function as kernel which will learn the last parameters w_i to infer the final value of the learning style. Figure 10 illustrate how the Active/reflective learning style pole is computed using our method. By this way, we can learn categorized learning styles regarding their learning features.

5 Data Analysis Measurement Analysis

5.1 Measurement Analysis

To measure the performance of models to infer learning styles, we have many metrics, the SIM similarity, the ACC (Accuracy) LACC (lowest Accuracy) and the Match [1–3, 9, 16] use the Receiver Operating Characteristic (ROC) to measure the performance of the model, this metrics is more stable indeed because the Accuracy of the model can be fake in cases where datasets are imbalanced. So, for this study we used the ROC curve and the Area Under the curve (AUC) as evaluation metric.

5.2 DBN-LIS Analysis

We used one unlabelled, and one labelled dataset to appreciate the model performance. For each dataset we split it into two (02) parts, one for the training data: 80% of the total dataset, and another one for the testing data (20%). The DBN-LIS used two (02) unsupervised hidden layers.

Unlabelled Dataset. After testing parameters on training data, we obtained the following best model parameters as illustrated in Table 4 for the Multi-Layer Perceptron and in Table 5 for the DBN approach:

The early stopping method avoids overfitting of the model, the validation data is introduced into the training stage to control its learning.

Performance. The evaluation was done on a multi-layer perceptron neural network. We used the area under the curve to evaluate the performance of the model.

Two algorithms have been used. The multilayer perceptron's (LSID-ANN) method with the backpropagation algorithm [5, 9], and DBN-LIS model. They gave good results, but from our modified Deep Belief Net we discover good improvement.

Table 4. Parameterization of ANN: multi-layer perceptron.

Dataset 1	#Features input layer	#Hidden neurons	Halting condition	Error threshold	# Epochs
Processing	11	10	Early stopping	0.01	100
Input	5	3	Early stopping	0.1	1000
Understanding	5	3	Early stopping	0.01	1000

Table 5. Learning parameters of the DBN

Dataset 1	#Visible units (input)	#Hidden neurons		# Epochs	Sampling step of contrastive divergence (CD)
		Layer 2	Layer 3		
Processing	33	66	16	700	1
Input	15	30	7	500	1
Understanding	15	30	7	500	1

Table 6. Comparative evaluation of the models on the unlabelled dataset.

Approach	Processing	Input	Understanding	Avg
LSID-ANN	92.25%	63.28%	86.96%	80.83%
DBN-LIS	99.9%	99.9%	99.9%	99.9%

Given the nature of unobserved data and learning traces in some learners, the LSID-ANN [9] model did not have enough strong predictive power on the Visual / Verbal and Sequential/Global learning styles, see Table 6. The DBN model considers a better analysis of the distributions of learning traces. Indeed, the first phase of the training process being unsupervised, removes from the learning style detection, the non-deterministic nature of the descriptors. The second phase of the DBN uses an SVM with radial basis function neuron, to predict the probability of observing a learning style, which is then applied a confidence level of 0.5 to infer the existence or not of the learning style.

Expert's Labelled Dataset. The labelled dataset uses learning traces of student and learning styles computed by the experts. We set the hyper parameters of the DBN-LIS using Bayesian optimization [19] which is faster than the Grid Search approach [3, 6] which is computer intensive. The best parameters to learn are given in Table 7. We selected the same learning features used by the authors on this dataset to ensure the effective for the modelling.

Table 7. Hyper parameters of DBN-LIS for labelled data

Datasets	#Visible units (input)	Hyperparameters					
		# Hidden neurons		#Epochs	Batch size	Learning rate	C
		Layer 2	Layer 3				
Processing	9	15	10	965	35	0.67	0.79
Perception	12	10	5	810	40	0.54	0.56
Input	6	15	15	560	10	0.87	0.89
Understanding	6	15	10	715	85	0.63	0.61

C is the SVM Self regularization parameter of the layer 4 in DBN-LIS. This parameter controls the overfitting process in the supervised learning of the layer4 on the network. This layer is the same size with the output of the unsupervised stage of the network.

After setting the parameters, we evaluate DBN-LIS on experts' dataset comparatively to J48 [11] and the Multilayer perceptron approach [1, 5, 9].

Table 8. DBN-LIS performance comparison

Model	Processing	Perception	Input	Understanding	mAUC
LSID-ANN	90.10%	87.70%	89.10%	76.90%	85.95%
J-48	91.76%	83.18%	90.16%	81.24%	86.59%
DBN-LIS	92.89 %	90.48%	95.55%	85.91%	91.21%

According to Fig. 11, the DBN-LIS method improves better than J-48 algorithm and the Multi-Layer Perceptron. Features on the SVM layer was refined by the unsupervised weight learning in the model; thus, permitting to have at this layer enriched learning traces representation for inference.

The mean AUC (mAUC) of **DBN-LIS** is high than previous models. This shows that DBN-LIS is effective for modeling learning styles. The perception dimension dropped down for **J-48** decision tree model due to the fact that the perception sub-dataset was unbalanced, a kind remarks is that Neural Nets can be strong and stable to the noise produce by the imbalanced data.

From the recent work [2, 7], Decision Trees approach are well appropriated in the aim of automatic detection of learning styles. The comparative evaluation in Fig. 11 confirmed this status, accordingly to the mAUC. Decision trees are more performant than natural Artificial Neural Net (ANN) [1, 9, 30]. The designed method DBN-LIS improved the existing approaches [9, 25] by modeling the prior approximation, which was experimental, and didn't analysis the distribution of learning patterns in the user model. Enhancing the user model thus strengthen the literature-based approach [25] to have better performance. This work also illustrates that using DBN-LIS, we can annotate

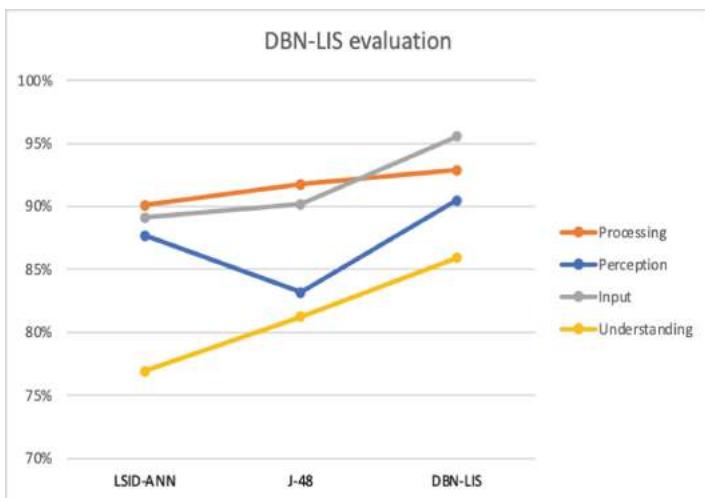


Fig. 11. Modeling of Felder and Silverman dimensions comparative evaluation.

efficiently students learning style basis on learning traces left on the system, without a regular usage of an expert.

6 Discussion

The results obtained at the end of this study show us that for an efficient calculation of learning styles improving the learning algorithm, we must find the most important learning indicators in order to generate a learning model, since in real cases, many learning traces in the system have a weak distribution that prevents enriching learning models. We also observe that the discretization of learning indicators promotes the modeling of learning traces distribution for DBN Neural networks. The use of DBN learning is much more relevant to the Artificial Neural Net, regarding the subjectivity and stochasticity nature of a learner behaviour on the e-learning platform. The methodology developed in this work is effective for the detection of learning styles, it could be further refined considering also the time parameter as the duration between learning interactions which was not included in the data modeling. The Moodle dataset used, permits to roll out our approach, pragmatical results were confirmed using Expert's labelled dataset using ILS and annotation of learners' preferences. In Table 6, comparatively to Table 8, the variation from 99.9% for mean AUC in Moodle dataset, to 91.21% to expert labelled dataset is due to the size of learning descriptors used in the LMS. The most DBN will have features to learn, the more the performance will be improved. After feature pre-processing, the Moodle dataset had more features than the Philippines dataset. Another important fact is the nature of the course, some courses can lead to week distribution on FSLSM, as exemplified on Table 2 where the perception dimension was not relevant in terms of learner's traces. In the case of the expert's dataset, the course is C++ learning course and involves sufficient representation of learner traces. The automatic ILS approach permits to identify and extract student with imbalanced learning

preferences Eq. 11, this is very important for the educationists in the sense that he will concentrate their efforts in adaptation strategy to student with strong preference learning style; however, the scale width of imbalance students is proportionally to the standard deviation of the prior estimation of learning style. One of the interests of DBN-LIS approach in spite the fact that it is non-deterministic by nature, is that it allowed the modeling of a gaussian distribution, this modeling process cover all the possible state of visible or discretize learning styles, then lead us to have the mean and right estimation of the posterior density function describing the learning style dimension. This study confirms our hypothesis in that, it is possible using our method to automatically infer student learning styles in spite of a questionnaire, through deep neural networks.

7 Conclusion

The personalization of learning in e-learning systems requires learner to be at the centre of their learning. In order to move from industrialized to personalized education, it is important that each training manager facilitate learning by providing learning resources aligned with learner learning styles. We have seen that theoretical models have had define categories of learners using learning styles that can be used to provide to learners rich learning activities, these methods admit some limitations by the subjectivity of questionnaires (ILS) dynamic detection techniques of learning styles have been proposed. Our study allowed us to develop a detection approach considering the stationary and non-deterministic aspect of learning styles. This approach has allowed us to study the distribution of learning indicators for the inference of learning styles. It has an advantage over existing approaches in that it doesn't require initialization of learning management system with some questionnaire strategy, moreover, it grants to reuse a model already learned in other online training courses using techniques such as model learning transfer (finetuning). The results we have obtained can be exploited in online training systems with the adaptable architecture proposed in this article to make learning activities dynamic and personalized. One of open issues in automatic learning styles modeling is the reliability of the detection model across different educational systems. The model built is highly coupled to the user model in the system, thus tend to be difficult to reuse in different learning systems. Find out how to make model reliable on several educational system, will be a good fit. Further works will consist of applying the DBN-LIS model on e-learning platforms to build adaptive user interface based on student learning styles, this will permit to generate an adaptive pedagogical model for learning content recommendation.

Annexure A

Algorithm 2: DBN-LIS learning procedure

Inputs:

Γ : Set of training set

lh : Number of hidden layers in the graph

η : Learning rate

Outputs

Θ : Model parameters.

1. Initialize $\Theta = \langle a, b, W \rangle$

$\text{Feat}_{\text{DBN}} = \{\}$

2. Foreach X in Γ

$disX = discretize(X)$

$\text{Feat}_{\text{DBN}} = \text{Feat}_{\text{DBN}} \cup disX$

3. Foreach layer l in $lh-1$

3.1 While not converged E_Θ

3.1.1 Select randomly v in Feat_{DBN} in training set

. Generate h' Sample of h given v (Positive stage)

$$P(h_i = 1|v) = \sigma(b_j + \sum_{i=1}^m w_{i,j}^l * v_i)$$

. reconstruct visible unit (Negative Stage)

$$P(h_i = 1|v) = \sigma(b_j + \sum_{i=1}^m w_{i,j}^l * v_i)$$

. Update model parameters

$$\Theta^l(t+1) = \Theta^l(t) + \eta(P(h_i = 1|v) - P(v_i = 1|h))$$

3.1.2 Compute cost reconstruction

$$E_\Theta = E(v, h)$$

4. Learn the weights of the last layer in the network with Support Vector Machines (SVM) using kernel radial basis function [33].

σ is an activation function, which can be the sigmoid or the Rectifier Linear Unit (ReLU) function.

References

1. Gomedé, E., de Barros, R.M., de Souza Mendes, L.: Use of deep multi-target prediction to identify learning styles. *Appl. Sci.* **10**(5), 1756 (2020). <https://doi.org/10.3390/app10051756>

2. Hmedna, B., El Mezouary, A., Baz, O.: A predictive model for the identification of learning styles in MOOC environments. *Clust. Comput.* **23**(2), 1303–1328 (2019). <https://doi.org/10.1007/s10586-019-02992-4>
3. Zhang, H., et al.: A learning style classification approach based on deep belief network for large-scale online education. *J. Cloud Comput.* **9**(1), 1–17 (2020). <https://doi.org/10.1186/s13677-020-00165-y>
4. Azzi, I., Jeghal, A., Radouane, A., Yahyaouy, A., Tairi, H.: A robust classification to predict learning styles in adaptive E-learning systems. *Educ. Inf. Technol.* **25**(1), 437–448 (2019). <https://doi.org/10.1007/s10639-019-09956-6>
5. Hasibuan, M.S., Nugroho, L.E., Santosa, P.I.: Model detecting learning styles with artificial neural network. *J. Technol. Sci. Educ.* **9**(1), 85–95 (2019). <https://doi.org/10.3926/jots.540>
6. Goodfellow, I., Bengio Y., Courville, A.: Deep Learning. MIT-Press (2018). <http://www.deeplearningbook.org>
7. Li, X., Abdul Rahman, S.S.: Students' learning style detection using tree augmented naive Bayes. *Roy. Soc. Open Sci.* **5**, 172108 (2018). <https://doi.org/10.1098/rsos.172108>
8. Sheeba T., Krishnan, R.: Prediction of student learning style using modified decision tree algorithm in e-learning system. In: Proceeding DSIT 2018 Proceedings of the 2018 International Conference on Data Science and Information Technology, pp. 85–90 (2018). ISBN: 978-1-4503-6521-5. <https://doi.org/10.1145/3239283.3239319>
9. Bernard, J., Chang, T., Popescu, E., Graf, S.: Learning style identifier: improving the precision of learning style identification through computational intelligence algorithms. *Expert Syst. Appl.* **75**, 94–108 (2017). <https://doi.org/10.1016/j.eswa.2017.01.021>
10. Elizabeth, D.: "Learn Moodle August 2016" anonymized data set. Moodle (2017). <http://research.moodle.net/158>
11. Maaliw III, R., Ballera, M.: Classification of learning styles in virtual learning environment using j48 decision tree. In: 14th International Conference on Cognition and Exploratory Learning in Digital Age (CELDA) (2017)
12. Sahid, D.S., Nugroho, L.E., Santosa, P.I.: Integrated stochastic and literate based driven approaches in learning style identification for personalized e-learning purpose. *Int. J. Adv. Sci. Eng. Inf. Technol.* **7**(5) (2017). <https://doi.org/10.18517/ijaseit.7.5.1745>
13. Amir, E.S., Sumadyo, A., Sensuse, D.I., Sucahyo, Y.G., Santoso, H.B.: Automatic detection of learning styles in learning management system by using literature-based method and support vector machine. In: 2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Malang, pp. 141–144 (2016). <https://doi.org/10.1109/ICA CSIS.2016.7872770>
14. Liyanage, M.P.P., Gunawardena, K.S.L., Hirakawa, M.: Detecting learning styles in learning management systems using data mining. *J. Inf. Process.* **24**(4), 740–749 (2016). <https://doi.org/10.2197/ipsjjip.24.740>
15. Feldman, J., Monteserein, A., Amandi, A.: Automatic detection of learning styles: state of the art. *Artif. Intell. Rev.* **44**(2), 157–186 (2015)
16. Abdullah, M., Alqahtani, A., Aljabri, J., Altowirgi, J., Fallatah, R.: Learning style classification based on student's behaviour in moodle learning management system. *Trans. Mach. Learn. Artif. Intell. (TMLAI)* **3**(1) (2015). <https://doi.org/10.14738/tmlai.31.868>
17. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**, 436–444 (2015). <https://doi.org/10.1038/nature14539>
18. Barber, M., Donnelly, K., Rizvi, S.: An avalanche is coming higher education and the revolution ahead: Institute for Public Policy Research (IPPR) (2013). <https://www.ippr.org/publications/an-avalanche-is-coming-higher-education-and-the-revolution-ahead>
19. Bergstra, J., Yamins, D., Cox, D.D.: Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In: Proceedings of the 30th

- International Conference on Machine Learning (ICML 2013), June 2013, pp. I-115–I-123 (2013)
- 20. Dorça, F.A., Lima, L.V., Fernandes, M.A., Lopes, C.R.: Comparing strategies for modeling students learning styles through reinforcement learning in adaptive and intelligent educational systems: an experimental analysis. *Expert Syst. Appl.* **40**(6), 2092–2101 (2013)
 - 21. Dung, P.Q., Florea, A.M.: An approach for detecting learning styles in learning management systems based on learners' behaviours. In: International Conference on Education and Management Innovation IPEDR, vol. 30. IACSIT Press (2012)
 - 22. Manouselis, N., Drachsler, H., Verbert, K., Duval, E. Recommender Systems for Learning. Springer Briefs in Electrical and Computer Engineering. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-1-4614-4361-2>
 - 23. Baldiris, S., Graf, S., Fabregat, R.: Dynamic user modeling and adaptation based on learning styles for supporting semi-automatic generation of IMS learning design. In: 2011 IEEE 11th International Conference on Advanced Learning Technologies, Athens, GA, pp. 218–220 (2011). <https://doi.org/10.1109/ICALT.2011.70>
 - 24. Hsu, C.-C., Wang, K.-T., Huang, Y.-M.: Modeling personalized learning styles in a web-based learning system. In: Pan, Z., Cheok, A.D., Müller, W., Zhang, X., Wong, K. (eds.) *Transactions on Edutainment IV*. LNCS, vol. 6250, pp. 12–21. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14484-4_2
 - 25. Graf, S., Kinshuk, Liu, T.-C.: Supporting teachers in identifying students' learning styles in learning management systems: an automatic student modeling approach. *Educ. Technol. Soc.* **12**(4), 3–14 (2009)
 - 26. Castro, F., Vellido, A., Nebot, A., Mugica, F.: Applying data mining techniques to e-learning problems. In: Jain, L., Tedman, R., Tedman, D. (eds.) *Evolution of Teaching and Learning Paradigms in Intelligent Environment*, vol. 62, pp. 183–221. Springer, Berlin (2007). https://doi.org/10.1007/978-3-540-71974-8_8
 - 27. David, A., Sergei, V.: k-means++: the advantages of careful seeding. In: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms, pp. 1027–1035 (2007). <https://doi.org/10.5555/1283383.1283494>
 - 28. Garcia, P., Amandi, A., Schiaffino, S., Campo, M.: Evaluating Bayesian net- works' precision for detecting students' learning styles. *Comput. Educ.* **49**(3), 794–808 (2007)
 - 29. Hinton, G.E., Osindero, S., Teh, Y.: A fast-learning algorithm for deep belief nets. *Neural Comput.* **18**, 1527–1554 (2006)
 - 30. Villaverde, J.E., Godoy, D., Amandi, A.: Learning styles' recognition in e-learning environments with feed-forward neural networks. *J. Comput. Assist. Learn.* **22**(3), 197–206 (2006)
 - 31. Brusilovsky, P., Peylo, C.: Adaptive and intelligent web-based educational systems. *Int. J. Artif. Intell. Educ.* **13**(2–4), 159–172 (2003)
 - 32. Felder, R.M., Solomon, B.A.: Index of learning styles questionnaire (1997). <http://www.engr.ncsu.edu/learningstyles/ilsweb.html>. Accessed 5 Feb 2009
 - 33. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**(3), 273–297 (1995)
 - 34. Felder, R.M., Silverman, L.K.: Learning and teaching styles in engineering education. *Eng. Educ.* **78**, 674–681 (1988)



Using Machine Learning to Identify Methods Violating Immutability

Tamás Borbély^(✉), Árpád János Wild, Balázs Pintér, and Tibor Gregorics

Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary
{pinter,gt}@inf.elte.hu

Abstract. A characteristic example of logic errors is the problem of poorly written Java methods that render immutable objects mutable by exposing member variables. These variables can then be used to modify or delete the underlying data regardless of class invariants, leading to serious problems such as data corruption or loss. In this paper, we offer a solution for identifying such functions by implementing a machine learning-based static code analyzer. We use a Recurrent Neural Network, trained on examples that were generated with a context-free grammar, to inspect and classify Java methods that return arrays of primitive or collections of non-primitive types. Every example can be classified as “exposes class”, “does not expose class”, or, if there is insufficient information, “cannot be decided”. The trained model is capable of classifying previously unseen code snippets with good accuracy.

Keywords: Mutability · Deep copy · Recurrent neural network · Static analyzer

1 Introduction

Making mistakes is unavoidable during software development and every one of them is a time bomb, a potential source of damage that can get triggered at any time. These can vary in seriousness from small, insignificant problems to even fatal errors. To have a better understanding of them and the way they behave they can be categorized. There are numerous options to choose from but the most common one, at least when it comes to compiled languages, is splitting them into *compilation*, *runtime*, and *logic* errors (bugs).

Compilation errors are the ones that occur while the compiler is creating an executable from the source code. After that is done and the application can be run all the errors that occur while the program is running are called runtime errors. Recognizing compilation errors is simple because one will not let the compiler finish. Runtime errors are less straightforward as some make the application crash while others may go unnoticed.

Logic errors are similar to the latter. These are parts in the source code that can be compiled and run but will not behave in the intended way. So when a program produces the expected output from a certain input it may or may not

contain logical errors. There is no simple way to tell the two cases apart as a bug is nothing more than a “correct” piece of code whose behaviour was meant to be different. Having said that, we can conclude that such an error is program-specific and a function that works fine (as it was intended) in one project may lead to errors in another one. In some cases, the culprit is not even a single piece of code or a function but the way other parts of the program interact with them.

Methods that expose class members make immutable objects mutable. What is a mutable object? An immutable (unchangeable) object is one whose state can not be changed after it is instantiated while a mutable (changeable) object’s state can be changed. To achieve immutability a class must have no setters and either it has to be properly encapsulated or can only have immutable fields. The two most significant advantages these objects offer, compared to mutable ones, are thread-safety and higher security.

Let’s suppose that we create a class called `A`, without setters. If `A` only has immutable fields that are final and cannot be changed regardless of their visibility, or if `A` does have mutable fields but they are encapsulated properly (the data is hidden in the object) then it will be immutable.

However, all the methods have to be written carefully as well. A poorly written getter method which does not deep copy the returned object will render `A` mutable in both cases. Due to the way Java handles its objects, getter functions need to be treated with special care if you want them to function duly. There are 8 primitive types in Java (byte, short, long, int, double, float, Boolean, char) that are passed and returned by value and everything else is considered a non-primitive data type (e.g. String, Arrays, other classes) which are passed and returned by reference. The very essence of getter functions is avoiding the exposure of members, so returning an object reference is almost the same as directly using members defined in the class. When a return parameter of a function is primitive or immutable everything will work properly, but as soon as it is a mutable object it has to be copied on return. Otherwise, it will be exposed. There are three basic copy conventions to choose from when it comes to returning an object:

- No copy: Returning a member of an object.
- Shallow copy: “duplicate as little as possible”, so not every element of an object will be copied, only its structure.
- Deep copy: Every element is duplicated and both the original and the new reference point to different memory locations.

The first two options both fail at fully separating the original and the new object, so a function must deep copy the returned parameter to avoid exposure. Deciding if an object has been successfully deep copied or not is not always easy as a lot of factors have to be taken into account. An object reference, in an object reference, in an object reference, etc. Depending on how deeply these references are nested into each other this process can become long and complicated.

In this paper, we offer a solution for a subset of this problem by finding and classifying functions that return arrays of primitive types or containers of primitive wrappers.

2 Related Work

Immutability has long been a debated topic. Although its safety and security preferences are indisputable, many other factors also have to be taken into consideration when building software. Weber [13] summarizes two previously published papers [3,4] about the usability and security impact of immutability. In semi-structured interviews with well-experienced software engineers, working on big projects with millions of lines of code, participants reported that the misuse of mutable objects was a significant source of bugs. One of them stated that all key data structures in their system's architecture were made immutable to ensure "... that an error or problem can never put the system into an undesirable state". The paper also mentions the lack of immutability support in certain languages that may decrease the desirability of immutability.

There are two state-of-the-art techniques to mutability analysis [5,8,9]. One of them is points-to analysis which is a static code analysis technique to decide which references can point to which storage locations. The algorithm builds a graph of where/what each pointer can point to. The other one is type inference which is a compile-time technique to fully or partially deduce the type of the variables in the code. Porat et al. [10] present a static analysis tool to identify mutability. It classifies components as "immutable", "mutable", or "undecided" by analyzing properties such as value or object accessibility. They described numerous situations where a component could lose its immutability. These can be difficult to find before compilation using traditional methods, so we use machine learning to offer a solution to part of this problem: identifying methods that return an object's field without properly copying it.

Our goal was to implement a static analyzer that is capable of recognizing such functions using machine learning without the need for explicit programming. Numerous authors found that inspecting source code using machine learning is a viable way of statically analyzing projects [6,11]. Barstad et al. [1] used neural networks to categorize applications as either "well written" or "badly written". They concluded that the implemented model carries out the desired task with high accuracy.

3 Method

Due to no dataset being available for this particular task we started by building one. A Context-Free Grammar in Python was created that is able to generate Java source code. It has the advantages of being both easy to use and very scalable. With little adjustments, the number of examples it is able to generate can increase from a couple of hundreds to hundreds of thousands.

A function can vary in length from 2–3 to even dozens of lines. Since getter methods exist to access private members of a class from outside they tend to be really short, provided they do not have extra functionality [7]. Even if they do, that information is not relevant to the copy mechanism as shown in Fig. 1. With that in mind, we generated examples only containing information related to copying.

```

1  public List<Long> getFileIds() {
2      try {
3          SleuthkitCase db = Case.getCurrentCaseThrows().getSleuthkitCase();
4          try (SleuthkitCase.CaseDbQuery queryResult =
5              db.executeQuery(getQuery(dataSourceId))) {
6              ResultSet resultSet = queryResult.getResultSet();
7              List<Long> fileIds = new ArrayList<>();
8              while (resultSet.next()) {
9                  fileIds.add(resultSet.getLong("obj_id"));
10             }
11         }
12     }
13 } catch (NoCurrentCaseException ex) {
14     throw new ExportRulesException("No current case", ex);
15 } catch (TskCoreException ex) {
16     throw new ExportRulesException("Error querying case database", ex);
17 } catch (SQLException ex) {
18     throw new ExportRulesException("Error processing result set", ex);
19 }
20 }
```

Fig. 1. The highlighted lines carry enough information to decide whether the returned object is successfully deep copied. In line 7 a new return type variable is declared and a new location in memory is assigned to it. In lines 8 and 9 a while loop iterates through *resultSet* and adds its next element to *fileIds*. These elements are long types and since long is immutable, the original data that is stored in the object cannot be changed through the returned *fileIds*.

3.1 Categories

To classify methods we created three different categories. These are the following:

- 0: It is clear that the function failed at creating a fully independent object (didn't copy the object at all or only shallow copied) so the original one is certainly mutable (Fig. 2)

```

1  Set<Integer> noCopy() {
2      return originalMember;
3  }
```

Fig. 2. Without copying anything, simply returning the original member

- 1: Successful deep copy, the returned array (or container) was duplicated with independent elements (Fig. 3)

```

1   Set<Integer> deepCopy() {
2       Set<Integer> out = new HashSet<Integer>();
3       for (Integer i : originalMember) {
4           out.add(i);
5       }
6       return out;
7   }

```

Fig. 3. Creating a new return type variable and properly copying all the data from the original member

- 2: Given information is insufficient, there are other function calls (methods, functions from other libraries, etc.) which might or might not deep copy (Fig. 4)

```

1   Set<Integer> functionCall() {
2       Set<Integer> out;
3       out = getIntList();
4       return out;
5   }

```

Fig. 4. Whether the memory location pointed by out is a new one or the same as the original member points to can not clearly be decided

These are the three classes we used to train the model. We also defined an order among these classes ((1) ; (2) ; (0)) because some examples may fall under more than one category. If a function has two return statements where one of them returns another function's value and the other returns a data member, choosing a class becomes ambiguous. It could be both type (0) or (2), so we use the order between them to decide. In practice, we generated examples which could be categorized as more than one category and labelled them with the category of highest precedence.

A function can be categorized as (1) only if no return value exposes the class. However, it is (0) if any of the return values exposes the class. If an example can not be categorized as (0) or (1) with a 100% certainty, then it falls under (2).

3.2 Architecture

To accurately capture the connection between tokens we used Bidirectional Long Short-Term Memory [12] (BLSTM) cells for the model (Fig. 5). Its first layer is an embedding layer (with an output dimension of 256) to prepare the data for the first Bidirectional LSTM layer that is followed by a Dropout layer, with a drop rate of 10%. These are followed by a second Bidirectional LSTM and a second 10% Dropout layer. Both of the BLSTM layers have 32 neurons. The

output layer of the model is a Dense layer with softmax activation and three neurons to represent our three classes.

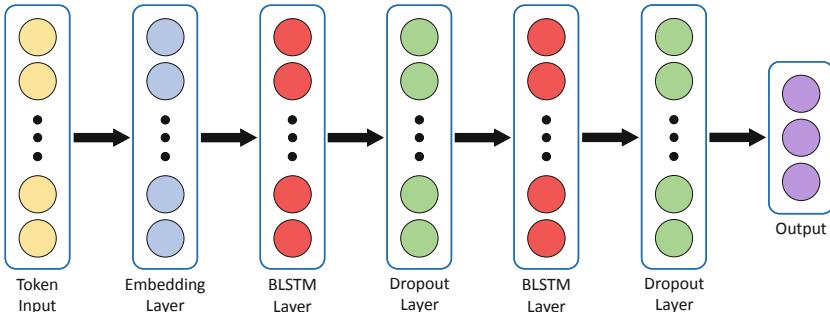


Fig. 5. Basic architecture of the model. It receives a 256-long token sequence as input which is then embedded into a 256 dimension vector space. There are 2 BLSTM layers with each containing 32 cells with a 10% dropout layer after both of them. Finally, the output of the model is a dense layer with 3 cells and softmax activation.

3.3 Training

A function was represented as a list of tokens, keeping their original order. An input length limit of 256 was defined and every example was adjusted to this parameter. This value was chosen as most of the functions were shorter. Every function that did not reach this limitation was expanded to this length by using pre padding method with a PADDING token (Fig. 6).

```
[PADDING, ..., PADDING, List, <, Byte, >, {, List, <, Byte, >, LaiBj, ;,
if, (, LDZOU, ==, LilxI, (, ), ), {, return, JGikJ, ;, }, LaiBj, =, Vakee, ;,
return, LaiBj, ;, }]
```

Fig. 6. Tokenized version of a single method (training example).

Most of the examples' lengths were between 10 and 256, padded to the maximum length of 256. To fit the model we used the categorical cross-entropy loss function and the Adam optimizer with a learning rate of 0.01. Due to the size of our data set, we found that few epochs (1–2) with a small batch size (16–32) were enough to achieve good performance on the test sets. We used a 75:20:5 split on the generated examples for training, validation, and testing.

4 Evaluation

In this section, we describe the method we used to create our example functions and the results our model achieved on the test sets.

4.1 Dataset

For every category, we generated approximately 5000 examples. Using the grammar (Sect. 4.2) made it possible to create slightly different but unique functions in large amounts. Then all the function variables were swapped with a random five-character-long string to prevent the model from deciding based on the presence/absence of a given variable.

Poorly written getter methods can be present in projects of arbitrary sizes. A more sensible way of categorizing them would be to separate the example functions by their complexity. Simpler examples are not longer than 8–10 lines and mostly contain information (return statement, declaration of a return type variable, for loop to copy items) that is strictly related to copying the returned object. Complex functions are longer than 10 lines and might contain a lot of additional information (arbitrary number of unrelated operations, exceptions, etc.).

4.2 Grammar

For each of the defined classes, a specific Context-Free Grammar (CFG) was built using the Python Natural Language Tool Kit (NLTK) [2]. Firstly, the base grammar was built to define a simple function structure, then three others to give every category their special behaviour. Every case was extended with if-else statements, simple assignments, function calls, etc. to let the model learn as many function structures as possible. An example of the rules is shown in Fig. 7.

To ensure that the generated methods are both syntactically (apart from the function’s declaration and signature) and semantically correct, we only used one type for the arrays and one type for the containers with fixed variable names. Later we swapped this fixed type with all the other primitive types and containers and changed all the variable names to a random, five-character-long string. The most significant advantage the grammar provided us with is that it could be expanded easily. After every successful validation evaluation, we could easily add more and more structure to our training set to widen the scale of the problems our model could deal with.

START	\rightarrow	TYPE '{' BODY '}' COLLECTION '{' LBODY '}'
COLLECTION	\rightarrow	' List<Byte> '
TYPE	\rightarrow	' byte[] '
ASSIGNMENT	\rightarrow	'' ASLEFT ASRIGHT
IF	\rightarrow	' if (' CONDITION ') {' IFBLOCK '}' ''
CONDITION	\rightarrow	VAR1 COMPARE VAR2 VAR1 EQUALITY BOOLEAN VAR1
COMPARE	\rightarrow	' > ' '<' EQUALITY
EQUALITY	\rightarrow	' ==' '!='
BOOLEAN	\rightarrow	' true ' ' false '

Fig. 7. A subset of the context-free grammar.

4.3 Results

To get a better picture on how the model performs we evaluated three test sets that vary in length and complexity. The first one was built from the 5% of the methods generated by the CFG that we did not use for training or validation (Table 1). The 100% accuracy on this set can be explained by the simplicity of the functions used.

Table 1. Test accuracy on CFG generated methods.

Category	Examples	Accuracy
Expose class (0)	304	100%
Doesn't expose class (1)	300	100%
Cannot be decided (2)	310	100%

The second test set also contained simple methods but these were not generated by the CFG (Table 2). We used functions we downloaded from GitHub and augmented them by changing variable names and parameter types. Most of the methods were less than 50 tokens long and did not contain a lot of additional code apart from copy related mechanics.

Table 2. Test accuracy on partially generated simple data set.

Category	Examples	Accuracy
Expose class (0)	510	100%
Doesn't expose class (1)	515	100%
Cannot be decided (2)	519	87.67%

Although this test set ended with a slightly lower accuracy as it could not always recognize examples that should have been classified as (2), the overall result is still 95%. This means that clean coded functions that do not contain a lot of additional functionality can be categorized with high precision.

However, it is not a realistic expectation that all class methods are that short and contain no extra functionality. Even if we are only talking about containers, they can be expanded, shortened, transformed: the overall length of an example, in terms of lines (or tokens), can be a lot greater. With that in mind, the main test set contained approximately 200 functions from various GitHub Java repositories with an average length of 60 tokens (Table 3).

Table 3. Test accuracy on the data set created from github projects.

Category	Examples	Accuracy
Expose class (0)	40	77.5%
Doesn't expose class (1)	74	94.59%
Cannot be decided (2)	70	60%

The overall accuracy of this test set was 78%. There are a couple of factors to take into consideration when analyzing this result. Apart from the fact these are longer functions, they may contain misleading information for the model such as copying an object which is unrelated to the return value. Although the algorithm recognizes these patterns, it can not always decide which parts (assignments, loops, declarations) should be ignored when predicting the output.

5 Conclusion

We presented a neural network that can decide whether methods that return arrays of primitive types or collections of primitive wrappers properly deep copy their values. This is especially important for immutable classes as breaking encapsulation can also violate their immutability.

We evaluated our approach on different test sets that contain code from real projects gathered from GitHub, using the model trained on the generated data. The model was able to generalize well: it successfully classified functions that are different from the training examples.

The trained network could be a useful tool to analyze whole Java projects and detect problematic methods. To achieve this an application would need to select the appropriate functions from the project and feed them to the model. Although we chose Java as the language of our datasets, the model can be easily modified to work with other similar programming languages such as C#.

Although the implemented Recurrent Neural Network proved to be an effective tool for analyzing source code, the defined categories could further be improved. Since the category “cannot be decided” suggests that the model is incapable of making a precise decision, it may reduce the overall usability. We would like to address this problem by making the algorithm able to extract and insert out of scope code, thus providing the model with more information to make a definite prediction.

References

1. Barstad, V., Goodwin, M., Gjøsæter, T.: Predicting source code quality with static analysis and machine learning. In: Norsk IKT-konferanse for forskning og utdanning (2014)
2. Bird, S., Loper, E., Klein, E.: Natural Language Processing with Python. O'Reilly Media Inc. (2009)

3. Coblenz, M., Nelson, W., Aldrich, J., Myers, B., Sunshine, J.: Glacier: transitive class immutability for java. In: 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 496–506. IEEE (2017)
4. Coblenz, M., Sunshine, J., Aldrich, J., Myers, B., Weber, S., Shull, F.: Exploring language support for immutability. In: 2016 IEEE/ACM 38th International Conference on Software Engineering (ICSE), pp. 736–747. IEEE (2016)
5. Huang, W., Milanova, A., Dietl, W., Ernst, M.D.: Reim & Reiminfer: checking and inference of reference immutability and method purity. ACM SIGPLAN Not. **47**(10), 879–896 (2012)
6. Mani, S., Sankaran, A., Aralikatte, R.: DeepTriage: exploring the effectiveness of deep learning for bug triaging. In: Proceedings of the ACM India Joint International Conference on Data Science and Management of Data, pp. 171–179 (2019)
7. Martin, R.C.: Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education, London (2009)
8. Milanova, A., Dong, Y.: Inference and checking of object immutability. In: Proceedings of the 13th International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools, pp. 1–12 (2016)
9. Milanova, A., Huang, W.: Dataflow and type-based formulations for reference immutability. In: International Workshop on Foundations of Object-Oriented Languages, p. 89. Citeseer (2012)
10. Porat, S., Biberstein, M., Koved, L., Mendelson, B.: Automatic detection of immutable fields in java. In: Proceedings of the 2000 conference of the Centre for Advanced Studies on Collaborative research, p. 10. IBM Press (2000)
11. Pradel, M., Sen, K.: Deep learning to find bugs. TU Darmstadt, Department of Computer Science (2017)
12. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
13. Weber, S., Coblenz, M., Myers, B., Aldrich, J., Sunshine, J.: Empirical studies on the security and usability impact of immutability. In: 2017 IEEE Cybersecurity Development (SecDev), pp. 50–53. IEEE (2017)



Predicting the Product Life Cycle of Songs on the Radio

How Record Labels can Manage Product Portfolios and Prioritise Artists by Using Machine Learning Techniques

O. F. Grooss^(✉), C. N. Holm, and R. A. Alphinus

Aarhus University BSS BTECH, 7400 Herning, Denmark
`{ofg, roal}@btech.au.dk`

Abstract. In terms of determining the success of a musical artist's song, there is a positive correlation of radio play success and music sales success. Therefore, being able to forecast the future plays of a song on the radio can serve as powerful risk management and product portfolio management tools for record labels and other stakeholders of a song. This research strives to predict the remaining product life cycle of a song on the radio after it has been played for one or two months. The best results were achieved using a k-d tree to calculate the songs the most similar to the test songs and use a Random Forest model to forecast radio plays. Accuracy of 82.78% and 83.44% was achieved for the two time periods, respectively. This explorative research leads to over 4500 test metrics to find the best combination of models and pre-processing techniques. Other algorithms tested were KNN, MLP, and CNN. The features only consist of *daily radio plays* and use no musical features.

Keywords: Hit song science · Product life cycle · Machine learning · Radio

1 Introduction

This paper seeks to investigate how to forecast the product life cycle of a song played on the radio, and it falls under the topic of Hit Song Science (HSS). HSS aims to predict if a song will become a hit. It is a subject explored within Music Information Retrieval (MIR). Hit Song Science assumes that hits have similarities, given that the right features are used [1]. Previous studies have used different approaches to predict whether songs will become a chart-topping hit or not [1–3]. Still, no paper has focused on predicting the product life cycle of songs played on the radio. Predicting future development in radio play serves two primary purposes. First, as there is a positive correlation of radio play success and music sales success, the success of a song on the radio indicates the future revenue for record labels, artists, and other stakeholders to the song [4]. Second, predicting whether a song will become a hit or not could reduce risk in investments made in newly signed artists [1]. The music organisation IFPI has estimated that record labels invest \$4.5 billion worldwide in talent discovery and development, including marketing. This makes it crucial to manage the product portfolio effectively. Information on how the

songs in the product portfolio will perform on the radio in the future allows the record label to prioritise songs and artists effectively. Therefore, this paper seeks to develop a machine learning model that is able to forecast the future number of plays for different types of songs. This paper assumes that the remaining product life cycle of a song can be predicted after one or two months by using the historical data from other songs that have completed their product life cycles (Fig. 1).

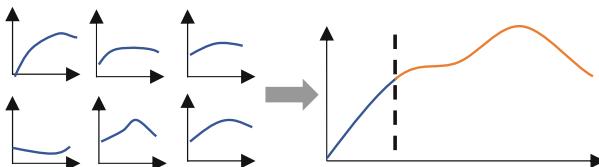


Fig. 1. Illustration of using previous life cycles of other songs to predict the remaining life cycle of a new song.

2 Dataset

As no dataset on historical radio play data is publicly available, the dataset had to be manually composed. The project behind this paper was conducted in collaboration with a company that specialises in monitoring radio stations worldwide and stores which songs are played where and when. Through their platform, it was possible to download historical data on 96 different songs to use for forecasting. Forty songs from top international artists were selected from Spotify playlists showing Top Hits in 2019 from artists with high popularity. Forty songs from German national artists were selected from Sing Deutsch, which is a website showing songs from German artists singing in German. This avoids a dataset filled with international artists with high popularity. Lastly, 16 songs were identified from artists that did not have high popularity before the release of their first big hit. This approach was used to attempt to represent the different types of songs in rotation on German radio. The data on every song were summarized into a total number of plays per day from Day 1, when the song was first detected on the radio, up until Day 160. The dataset represents the aligned product life cycles for each song; the rows are the samples and the columns are the features representing plays per day from Day 1–160.

3 Clustering

Initially, the dataset was submitted to clustering with a K-means clustering algorithm, and the Euclidean distance was used to calculate the distance between the songs in the dataset. The Euclidean distance measures the distance by using the root of the sum of each delta distance squared [5]. From that, it uses Pythagoras' approach of calculating the hypotenuse of a triangle. Regardless of which data clustering technique is used, the

Euclidean distance is a mathematical principle that follows through and can be written as (1).

$$d^2(x, y) = \sum_{j=1}^m (x_j - y_j)^2 = \|x - y\|_2^2 \quad (1)$$

The clustering was repeated using $k = 1, 2 \dots 20$ to establish the optimal number of clusters. The Elbow method pointed to optimal cluster sizes of two or five. When using $k = 2$, the algorithm split the dataset into one group of songs that reached above 700 daily plays and another group of songs that did not reach 700 daily plays. Using $k = 5$, the algorithm also used how many plays the songs reached, but at the same time, it used the time dimension of how fast the songs reached their maximum daily plays. It split the dataset into three sizes in terms of how big of a hit the songs were. This was labelled as follows: *big hit*, *medium hit*, and *small- or non-hit*. The time dimension was used on the *big hit* and *medium hit* songs, and they were split into two categories as *rapidly* or *slowly* reaching their maximum daily plays. As these clusters utilise the time dimension in the data, $k = 5$ was chosen. The clusters were labelled, as seen in Fig. 2.

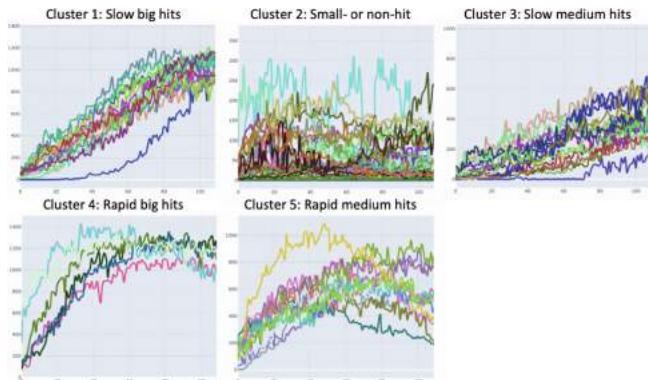


Fig. 2. K-Means clustering using $k = 5$

Two test songs were chosen from each cluster to create a representative test set. As seen in Fig. 3, the ten test songs represent different product life cycles of songs on the radio.

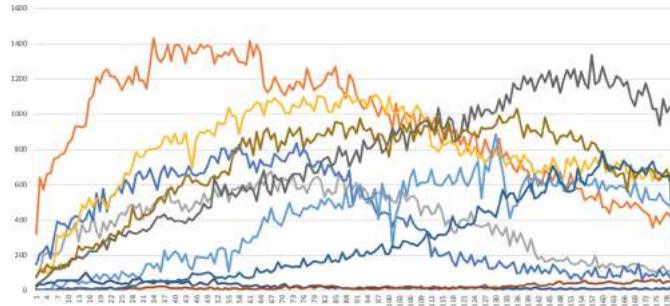


Fig. 3. The radio plays of the ten test songs selected

4 Designs

Two time periods were established: *Time Period 1*, where 28 days of historical data are used to predict the daily plays for the next 84 days; and *Time Period 2*, where 56 days of historical data are used to predict the daily plays for the next 84 days. To do this, we proposed three approaches. In *Design A*, the algorithms use the entire dataset to make forecasts. This means that when a new song (marked with a big X on Fig. 4) needs to be predicted, the algorithm will train on all of the other data points in the dataset. After doing this, the new data point will be compared to both similar and dissimilar data points that are indicated by the distance between the new data point and the training set data points. In *Design B*, the models train on the clustered dataset. This means that the model will only train on songs that are somewhat similar to the test song. Training on only similar datapoints has proven effective in other contexts [6]. However, there are complications to this approach. If the new data point turns out to be an outlier within the cluster, it will end up training on data points in the other end of that cluster even though there are data points from other clusters that are closer to it. Another challenge with this approach is that it requires a classifier that is capable of correctly assigning new songs to the correct cluster. Lastly, *Design C* uses similar data points relative to the new data point. In this approach, the new data point is only trained on data that looks like itself and ignores all other data points. This approach might seem very similar to the previous one, but the main difference here lies in how similarity is identified. In the previous approach, the similarity is identified by measuring the distance between every single point and clustering the points based on points with short distances between them. In this approach, several nearest neighbours to the new data point are identified, and the prediction is made from these points. One of the algorithms utilising this concept is the K-Nearest Neighbours (KNN) algorithm. An essential part of this algorithm is the *k-d tree algorithm*, which is responsible for identifying the nearest neighbours from which the KNN model can make predictions [7]. The k-d tree will therefore be used in combination with other machine learning models where the k-d tree algorithm will create datasets for the models to train on.

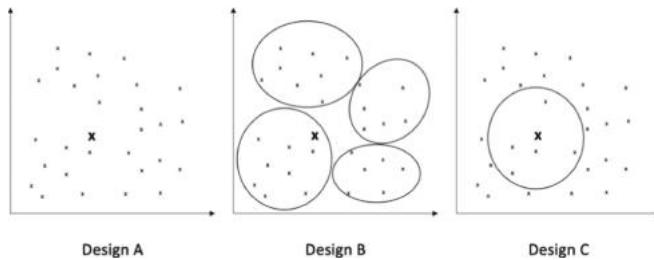


Fig. 4. Conceptual visualisation of the three approaches

5 Models

To select appropriate models for this problem, the literature on Hit Song Science and time series problems was reviewed.

5.1 KNN

KNN is considered to be one of the simplest machine learning algorithms. However, it has proven extremely effective because it can handle broad diversity in the data [8]. Despite its simplicity, it has proven effective on highly non-linear and complex time series problems [9]. When forecasting, the KNN algorithm searches its feature space to identify the k number of training instances where the features are nearest to the features of the sample that needs to be predicted [10]. Then the dependent variable is predicted as a weighted average of these k nearest output observations in the data [11]. As KNN regression predicts the average target values from the historical data in a time series, the algorithm is not able to predict values outside the range of the existing observations, which is why it does not handle outliers well. There are several options to choose from when calculating the distance to the neighbours, choosing the number of neighbours to use, and calculating the weight [11]. The variable k can be chosen by taking the square root of the number of training instances or by using an optimisation tool. KNN regression is defined in (2):

$$\hat{m}_k(x) = n^{-1} \sum_{i=1}^n W_{ki}(x) Y_i \quad (2)$$

where W_{ki} is the weight, Y_i is the local averages of the output variables, and x is the neighbourhood (bandwidth) [12]. The observations can be weighted equally (uniform) or by *distance*, where the closest neighbours have the most impact. KNN can be used for time series forecasting that contains repetitive patterns by finding the previous patterns that are similar to the pattern of the current series and using these patterns to predict the value. Data are often normalised for KNN, but as all observations in time series are on the same scale, normalisation is not necessary on time series problems for KNN.

5.2 Random Forest

Random Forest (RF) is an ensemble method using several decision trees and applying bagging techniques. It has proven effective when working with small datasets [13]. RF

adds a random subset selection of features as well [14]. In regression problems, RF uses random regression trees and aggregates all of the trees' predictions to create a final output prediction. This means that instead of calculating Information Gain and Gini Index to determine the best features for separation, RF uses random subsets of features and then calculates the best of these features for separation. RF builds many smaller trees (weak learners), which reduces overfitting and improves generalisation ability [15]. It is a good choice when the number of variables exceeds the number of samples [16].

RF is a collection of M trees. At the query point x , the prediction can be denoted as $m_n(x; \Theta_j, D_n)$, where $\Theta_1, \dots, \Theta_M$ are random variables and D_n is a given training sample $((X_1, Y_1), \dots, (X_n, Y_n))$ [16]. Mathematically, the j -th tree can be written as shown in (3):

$$m_{M,n}(x; \Theta_j, D_n) = \sum_{i \in D_n^*(\Theta_j)} \frac{1_{X_i \in A_n(x; \Theta_j, D_n)} Y_i}{N_n(x; \Theta_j, D_n)} \quad (3)$$

where $D_n^*(\Theta_j)$ is the selected datapoints, $A_n(x; \Theta_j, D_n)$ is the cell that contains x , and $N_n(x; \Theta_j, D_n)$ is the number of datapoints that fall into the cell. The average of all the predictions from the trees can be written as in (4).

$$m_{M,n}(x; \Theta_1, \dots, \Theta_M, D_n) = \frac{1}{M} \sum_{j=1}^M m_n(x; \Theta_j, D_n) \quad (4)$$

5.3 MLP

Multi-Layer Perceptron (MLP) is one of the simplest neural networks, and it is included due to the small dataset, as the performance is more controlled by the weights than the network [17]. The network consists of one input layer, one or several hidden layers, and one output layer. The number of neurons in the input layer depends on the shape of the training data. The hidden layers take these input values, pass them through the neurons of the hidden layers, and calculate an output with an activation function at each neuron. This is known as *feeding forward*. At the end of the chain, the output layer produces the final result from what was given from the hidden layer. Figure 5 shows the structure of the MLP used in this paper.

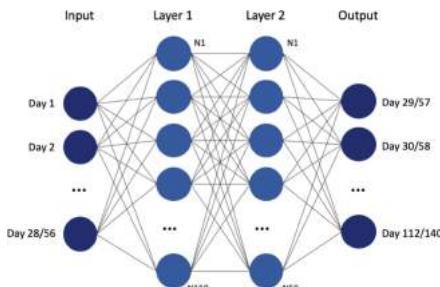


Fig. 5. Visualisation of MLP

The values from an input layer can be denoted as x_i and an added weight w_i , which is the strength of the connection between the neurons. The output is denoted by z , and it is calculated by adding bias b to the dot product [18], which can be written as (5).

$$z = x \cdot w + b \quad (5)$$

The rectified linear unit (**ReLU**) activation function is one of the most widely used activation functions for DL architectures [19]. ReLU uses a threshold where values less than zero are set to zero, thus solving the vanishing gradient problem [18]. The activation function is given by (6).

$$f(x) = \max(0, x) = \begin{cases} x_i, & \text{if } x_i \geq 0 \\ 0, & \text{if } x_i < 0 \end{cases} \quad (6)$$

With the predicted outputs, denoted \hat{y}_i , from the activation functions, the difference to the true value y_i , can be calculated. The error of the entire dataset is calculated with the cost function, and the goal is to find the best weights and biases for the perceptrons. The relationship between the cost function and the weights and biases can be found by calculating the gradient of the cost function with respect to the weights and biases. This calculation method is called *backpropagation*. The backpropagation updates the weights for each layer, starting with the weights for the output layer and working its way backwards through the hidden layers. It calculates the derivative of the gradient with respect to each of the weights in the network. The optimiser does this over and over until it has found the minimised loss [20]. In neural networks, the parts are dependent on each other. The loss function C_0 depends on the activation output $a_j^{(l)}$, which depends on the input node $z_j^{(l)}$, which depends on the weight $w_{jk}^{(l)}$. These dependencies enable a calculation to find the best weights by taking the partial derivative of the error/cost function. This is called the chain rule [21], and in this case it is given by (7).

$$\frac{\partial C}{\partial w_{ij}^{(L)}} = \left(\frac{\partial C}{\partial a_j^{(L)}} \right) \left(\frac{\partial a_j^{(L)}}{\partial z_j^{(L)}} \right) \left(\frac{\partial z_j^{(L)}}{\partial w_{ij}^{(L)}} \right) \quad (7)$$

5.4 CNN

Convolutional Neural Networks (CNN) is one of the most notable deep learning algorithms, as the multiple layers are trained to be more robust. CNNs are inspired by the natural vision perceptual system and are most often found in computer vision applications. CNN generally consists of three main neural layers: a convolutional layer, a pooling layer, and a fully connected layer [19]. Convolutional layers consist of filters and feature maps. Filters are basically the neurons of the convolutional layer and are square matrices of weights that are slid across each picture in order to extract features. Pooling layers are added after convolutional layers. The pooling layer basically does the same as a convolutional layer by scanning the previous layer with a filter that is defined with size and stride [22]. However, instead of extracting the dot product of the scanned

area, a max-pooling layer extracts the maximum value and outputs it to a feature map [19, 22]. By doing this, the max-pooling layer extracts the most significant feature read by the filter. A fully-connected layer connects all neurons in one layer to all neurons in the next layer. In a classification problem, the objective for a fully connected layer is to take the output from the convolution and pooling process and decide to which class the instance belongs. It receives a flattened vector from the convolution/pooling layers, which contains values that each represents the probability of a feature belonging to a class. Each neuron at the output layer (the end of a fully-connected layer) represents a class. The fully connected layer has its own backpropagation, where weights are updated for the neurons. In the end, the classification happens as each neuron votes on a label, and the highest voted label wins [23].

6 Findings

The described models were all applied to the dataset in order to establish the best performing model (Table 1). In Design A, the best performing model was the MLP on Time Period 1 and CNN on Time Period 2. In Design B, the overall performance of all models increased. This is believed to be caused by the models only training on somewhat similar data. Here, MLP was again the best performer on Time Period 1 and KNN on Time Period 2. However, as Design B assumes that each song can be classified as belonging to one of the clusters, a classifier is required. This classifier could not be validated in this research because it showed poor results. Therefore, the best performing approach is Design C, where RF was the best performing model in both time periods. The optimal number of neighbours was established to be $k=11$. This proves that training on similar data increases accuracy.

Table 1. Accuracy achieved for each design presented

	Time period	Model	Accuracy
Design A	Time period 1	MLP	76.35%
	Time period 2	CNN	76.24%
Design B	Time period 1	MLP	88.31%
	Time period 2	KNN	86.31%
Design C	Time period 1	RF	83.44%
	Time period 2	RF	82.78%

7 Discussion

We selected ten test songs based on the clusters to represent different types of songs. Therefore, any error in the clustering process is likely to have had an impact on the forecasting results. To validate the results, we chose ten new validation songs and compared

the results. We were not able to test on more songs due to the small size of the dataset and due to the dataset being imbalanced in terms of clusters. We acknowledge the limitation of claiming accuracy of a model when only validating on 20 samples. More research is needed to establish the true generalisation ability of the models when predicting radio plays. This research is merely explorative research for predicting radio plays. The accuracy is lower in Time Period 2 with 56 days input than Time Period 1 with 28 days of input, which was unexpected. We believe it is a harder task to forecast Time Period 2 due to the divergence between songs as they mature in their product life cycle. On the other hand, Design C handles this well, as it keeps its accuracy for Time Period 2 close to Time Period 1. Here, the k-d tree can calculate a better neighbourhood with more historical data, which is why good accuracy is still achieved on Time Period 2. When examining the results in terms of clusters, all algorithms generally performed well on Clusters 1, 4, and 5 but struggled in distinguishing between Cluster 2 and Cluster 3. This intuitively makes sense as the historical data within one and two months are very similar for these two clusters. To predict Cluster 3 songs, more historical data are needed. The rise in daily plays has a different starting point for these songs, making it hard to identify how much historical data are needed. To predict these types of songs, we propose using a dynamic forecasting period relative to the new instance. An upwards trend in daily plays needs to be detected to activate the model, recalculate the neighbours to train on, and create the forecast. Such an alert system might be able to detect Cluster 3 songs and successfully forecast them. We expect that more domain-specific features can help the forecasting of radio plays. The clustering of songs showed that popular artists often reach their peak in radio plays earlier than artists with low popularity. For this reason, we suggest constructing a dataset with popularity over time. This could be fed to the machine learning model as a vector to measure the correlation between radio plays and popularity.

8 Future Work

The findings are a result of structuring the data as univariate input and univariate output. If the success of a song depends on certain popular radio stations putting the song on rotation, then complex models might be able to spot the relationship between daily plays on these radio stations and the total daily plays in the future. When combining plays per radio station with plays per day, a multivariate input could be constructed. The output could be structured as univariate to predict daily plays. It could also be structured as multivariate to predict daily plays per radio station. We acknowledge that early indicators of radio success for artists might exist in other sources such as Spotify and TikTok. Combining data from streaming, social media, and radio could result in higher forecasting accuracy.

References

1. Middlebrook, K., Sheik, K.: Song Hit prediction: predicting billboard hits using spotify data, pp. 1–6, August 2019. <http://arxiv.org/abs/1908.08609>

2. Cibils, C., Meza, Z., Ramel, G.: Predicting a song's path through the billboard hot 100. Stanford Univ. Calif., pp. 1–6 (2015). cs229.stanford.edu/proj2015/012_report.pdf
3. Herremans, D., Martens, D., Sörensen, K.: Dance hit song prediction. *J. New Music Res.* **43**(3), 291–302 (2014). <https://doi.org/10.1080/09298215.2014.881888>
4. Dewan, S., Ramaprasad, J.: Social media, traditional media, and music sales. *MIS Q.* **38**(1), 101–121 (2014). <https://doi.org/10.25300/MISQ/2014/38.1.05>
5. Tibshirani, R., Walther, G., Hastie, T.: Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. B* **63**, 411–423 (2001)
6. Hu, K., Acimovic, J., Erize, F., Thomas, D.J., Van Mieghem, J.A.: Forecasting new product life cycle curves: practical approach and empirical analysis. *Manuf. Serv. Oper. Manag.* **21**(1), 66–85 (2019). <https://doi.org/10.1287/msom.2017.0691>
7. Chen, Y., et al.: Fast neighbor search by using revised k-d tree. *Inf. Sci. (Ny)* **472**, 145–162 (2019). <https://doi.org/10.1016/j.ins.2018.09.012>
8. Susto, G.A., Schirru, A., Pampuri, S., McLoone, S., Beghi, A.: Machine learning for predictive maintenance: a multiple classifier approach. *IEEE Trans. Ind. Informatics* **11**(3), 812–820 (2015). <https://doi.org/10.1109/TII.2014.2349359>
9. Parmezan, A.R.S., Souza, V.M.A., Batista, G.E.A.P.A.: Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Inf. Sci. (Ny)* **484**(February), 302–337 (2019). <https://doi.org/10.1016/j.ins.2019.01.076>
10. Chu, Y., Coimbra, C.F.M.: Short-term probabilistic forecasts for direct normal irradiance. *Renew. Energy* **101**, 526–536 (2017). <https://doi.org/10.1016/j.renene.2016.09.012>
11. Haara, A., Kangas, A.: Comparing K nearest neighbours methods and linear regression—is there reason to select one over the other? *Math. Comput. For. Nat. Sci.* **4**(1), 50–65 (2012)
12. Beckett, S.: Nonparametric regression: Kernel, WARP, ad k-NN estimators, no. May (2014)
13. Tyralis, H., Papacharalampous, G., Langousis, A.: A brief review of random forests for water scientists and practitioners and their recent history in water resources. *Water* **11**(5), 910 (2019). <https://doi.org/10.3390/w11050910>
14. Zhang, C., Ma, Y.: Ensemble Machine Learning. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-1-4419-9326-7>
15. Kaparthi, S., Bumblauskas, D.: Designing predictive maintenance systems using decision tree-based machine learning techniques. *Int. J. Qual. Reliab. Manag.* **37**(4), 659–686 (2020). <https://doi.org/10.1108/IJQRM-04-2019-0131>
16. Biau, Gérard., Scornet, E.: A random forest guided tour. *TEST* **25**(2), 197–227 (2016). <https://doi.org/10.1007/s11749-016-0481-7>
17. Ingrassia, S., Morlini, I.: Neural network modeling for small datasets. *Technometrics* **47**(3), 297–311 (2005). <https://doi.org/10.1198/004017005000000058>
18. Nwankpa, C., Ijomah, W., Gachagan, A., Marshall, S.: Activation functions: comparison of trends in practice and research for deep learning, pp. 1–20, November 2018. <http://arxiv.org/abs/1811.03378>
19. Gu, J., et al.: Recent advances in convolutional neural networks. *Pattern Recognit.* **77**, 354–377 (2018). <https://doi.org/10.1016/j.patcog.2017.10.013>
20. Werbos, P.J.: Backpropagation through time: what it does and how to do it. *Proc. IEEE* **78**(10), 1550–1560 (1990). <https://doi.org/10.1109/5.58337>
21. Zhang, A., Lopton, Z.C., Li, M., Smola, A.J.: Dive Into Deep Learning (2020)
22. Wang, K., et al.: Multiple convolutional neural networks for multivariate time series prediction. *Neurocomputing* **360**, 107–119 (2019). <https://doi.org/10.1016/j.neucom.2019.05.023>
23. Le Guennec, A., Malinowski, S., Tavenard, R.: Data augmentation for time series classification using convolutional neural networks. In: ECML/PKDD Workshop on Advanced Analytics and Learning Temporal Data (2016)



Hierarchical Roofline Performance Analysis for Deep Learning Applications

Charlene Yang¹, Yunsong Wang¹⁽⁾, Thorsten Kurth², Steven Farrell¹, and Samuel Williams¹

¹ Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

{cjyang,yunsongwang,sfarrell,swwilliams}@lbl.gov

² NVIDIA Corporation, Santa Clara, CA 95051, USA

tkurth@nvidia.com

Abstract. This paper presents a practical methodology for collecting performance data necessary to conduct hierarchical Roofline analysis on NVIDIA GPUs. It discusses the extension of the Empirical Roofline Toolkit for broader support of a range of data precisions and Tensor Core support and introduces a Nsight Compute-based method to accurately collect application performance information. This methodology allows for automated machine characterization and application characterization for Roofline analysis across the entire memory hierarchy on NVIDIA GPUs, and it is validated by a complex deep learning application used for climate image segmentation. We use two versions of the code, in TensorFlow and PyTorch respectively, to demonstrate the use and effectiveness of this methodology. We highlight how the application utilizes the compute and memory capabilities on the GPU and how the implementation and performance differ in two deep learning frameworks.

Keywords: Roofline model · Performance analysis · Memory hierarchy · NVIDIA GPUs · Deep learning · Image segmentation

1 Introduction

The Roofline model [34] is an intuitive performance model that can offer valuable insights into application performance, performance bottlenecks, and possible optimization opportunities. Its capability to extract the key computational characteristics and abstract away the complexity of modern computer architectures has gained its popularity in recent years in both traditional high-performance computing (HPC) and machine learning. Roofline is a throughput-oriented model centered around the interplay of computational capabilities, memory bandwidth, and data locality. Data locality is expressed as the arithmetic intensity (AI), the reuse of data once it is being loaded from memory, and it is commonly calculated as the ratio of the floating-point operations performed to the data movement, i.e. FLOPs per byte. The sustained performance (GFLOP/s) is then bound by two terms:

$$\text{GFLOP/s} \leq \min \begin{cases} \text{Peak GFLOP/s} \\ \text{Peak GB/s} \times \text{Arithmetic Intensity} \end{cases} \quad (1)$$

The Roofline model conventionally only focuses on one level in the memory hierarchy, but this has been extended in recent years to the full memory system to help understand cache reuse and data locality and provide additional insights into code performance. To facilitate the Roofline study, many tools and workflows have sprung to life, for example, the Empirical Roofline Toolkit (ERT) developed at the Lawrence Berkeley National Laboratory, for more accurate machine characterization [6, 38], and other tools, methodologies, and workflows for more streamlined application performance data collection in [8, 30, 37, 39]. A range of studies have also been conducted on the application of Roofline in both traditional HPC [18, 20, 21, 26, 35, 39] and Machine Learning [24, 33, 39], and the extension and refinement of the model to other related topics such as instruction Roofline [19], time-based Roofline [33], Roofline scaling trajectories [23], performance portability analysis based on Roofline [38], and power and energy Roofline [17, 29].

Deep learning has become one of the most dominant tools in areas such as pattern recognition, object detection, image segmentation, and language processing [22, 28], and its training or inference process usually takes a long time and requires significant computational resources. To tackle this problem, many innovative methods have been proposed [25, 27] to scale up such applications, and in this paper, we will focus on the Roofline-based performance modeling to analyze and examine how well various deep learning frameworks are utilizing the different aspects of the computer architecture, especially NVIDIA GPUs.

We will propose a practical methodology for collecting necessary performance data to conduct hierarchical Roofline analysis on NVIDIA GPUs. There are two components to this methodology, machine characterization using the Empirical Roofline Toolkit (ERT) [6] and application characterization using Nsight Compute [9]. We will discuss the extension of ERT for support on multiple data precisions and Tensor Core operations, and the Nsight Compute metrics used to measure application performance such as the run time, sustained throughput, and data movement across the entire memory hierarchy. This methodology then will be validated by a state-of-the-art deep learning application, DeepCAM [27], in climate image segmentation, to demonstrate its effectiveness in application analysis. Two versions of the code will be examined, in TensorFlow and PyTorch respectively, and some insights will be highlighted on how deep learning applications, in general, utilize the compute/memory capabilities on NVIDIA GPUs and how the two deep learning frameworks, TensorFlow and PyTorch, can differ in implementation and performance.

2 Methodologies

In this section, we will discuss the extension work done on the Empirical Roofline Toolkit (ERT) in order to support multiple data precisions (such as FP16) and

Tensor Core operations on NVIDIA GPUs, and the set of metrics in `Nsight Compute` that can be used to measure application performance such as run time, sustained throughput and data movement at different levels of the memory hierarchy. These two components together comprise the complete data collection methodology for machine and application characterization in a hierarchical Roofline analysis on NVIDIA GPUs.

2.1 ERT Extensions for Machine Characterization

The Empirical Roofline Toolkit (ERT) [6] is developed and maintained by the Lawrence Berkeley National Laboratory. It consists of micro-kernels that are finely tuned to test the various aspects of computer architecture such as memory bandwidth and compute throughput. Compared to theoretical values or marketing numbers from vendors, this provides a more accurate understanding of the architecture's capability in real programming environments with real power, thermal constraints, and programming models.

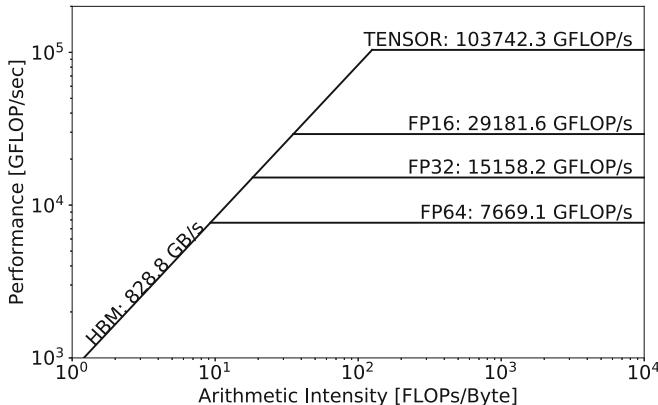


Fig. 1. Roofline graph generated by empirical results for V100 GPU including the new reduced-precision and tensor core ceilings.

ERT is essentially a Python script that wraps around a range of micro-kernels written in C++ and parallelized with various programming models on different architectures. For example, OpenMP and MPI are used on Intel CPUs, CUDA is used on NVIDIA GPUs, and more micro-kernels are currently being added to support AMD architectures, IBM Power processors, and Intel GPUs. These micro-kernels are specifically tuned to test different aspects of the architecture and provide an upper bound for real-life applications on them, i.e. if such kernels can not reach certain performance, there is almost no hope for large complex applications in real life to achieve it.

The ERT prior to this paper only supports double precision (FP64) performance characterization and in this section, we will detail how we have extended

it to support single-precision (FP32), and half-precision (FP16), as well as Tensor Core operations on NVIDIA GPUs. The resultant Roofline ceilings are shown in Fig. 1, with 7.7 TFLOP/s for FP64, 15.2 TFLOP/s for FP32, 29.2 TFLOP/s for FP16 on the CUDA core, and 103.7 TFLOP/s on the Tensor Core, on V100 GPUs.

Single-Precision (FP32) and Half-Precision (FP16). The original ERT is written in C and only supports double precision (FP64) measurements. While this can be easily extended to single-precision (FP32) by replacing ‘double’ by ‘single’ in the code, it requires work to support half-precision (FP16). For maintainability and future extensibility purposes, we have rewritten ERT in C++ and leverage C++ templates to support multiple data types.

Table 1. FP16 performance on the CUDA core on V100 GPUs (the higher, the better)

Version	Implementation	Performance (TFLOP/s)
v1	Naive	15.421
v2	Replace <i>half</i> with <i>half2</i>	20.142
v3	<i>uint32_t</i> for indexing	28.152
v4	Inline intermediate variables	28.376
v5	<i>uint32_t</i> only	29.182

For FP32, we have easily obtained 15.2 TFLOP/s peak performance, which is within 5% of the advertised 15.7 TFLOP/s performance [31]. For FP16 (on the CUDA Core), some performance tuning is required as detailed in Table 1. The naive implementation (v1) simply passes *half* as the data type to the templated functions and that resulted in a similar performance to the FP32 precision’s, 15.4 TFLOP/s. This is because V100s do not support FP16 directly on the CUDA Core [31] and each FP16 operation is essentially executed as an FP32 operation (i.e. going through the same pipeline). To efficiently perform FP16 operations (even though utilizing the Tensor Core would be a good option), on the CUDA Core, a vector type *half2* can be used to pack two FP16 values together to one FP32 register and be executed in one FP32 instruction. In ERT, we have implemented this using intrinsic functions and obtained an improved performance of 20.1 TFLOP/s (v2) in Table 1. In real life, it is not practical to implement large scale applications in intrinsics but yet the implementation is an attempt to push the Roofline ceiling as high as we possibly can.

The rest three versions v3-v5 in Table 1 are a series of optimizations that have proved to be beneficial to the development of ERT and are expected to be largely helpful to real-life applications and their performance tuning as well. Out of the three, replacing *uint64_t* indexing variables with the *uint32_t* data type has proven to bring the most performance gain, from 20.1 TFLOP/s to

28.2 TFLOP/s. This is due to the fact that V100s only support INT32 integer operations on the hardware level and that there is constant type conversion between *uint64_t* and *uint32_t* for the second version of ERT (v2). With the inlining of intermediate variables in v4 and conversion of all integers to *uint32_t* in v5, the FP16 CUDA Core performance of ERT has been brought on par to the theoretical peak with 29.2 TFLOP/s in Fig. 1.

Tensor Core. NVIDIA Tensor Cores are designed to accelerate matrix-matrix multiplication operations, which represent the mathematical nature of many deep learning workloads, for example, convolutional neural networks (CNNs). They operate on 4×4 matrices and can perform the following matrix multiplication and accumulation extremely efficiently:

$$D = A \times B + C \quad (2)$$

where A and B are matrices in FP16, and C and D are matrices in either FP16 or FP32. V100 has 80 SMs and 8 tensor cores per SM, and at 1.312 GHz clock frequency, its theoretical Tensor Core peak can be calculated as:

$$80 \times 8 \times 1.312 \times 4^3 \times 2 = 107.479 \text{ TFLOP/s} \quad (3)$$

To stress-test the Tensor Cores on V100, we have implemented ERT based on general matrix-matrix multiplications (GEMMs), where α and β are constant coefficients:

$$D = \alpha * A \times B + \beta * C \quad (4)$$

In general, there are two ways to program on Tensor Core, using the WMMA (Warp Matrix Multiply Accumulate) API in CUDA [2], or libraries such as cuBLAS [3] and cuDNN [16]. The *nvcuda::wmma* namespace in CUDA provides specialized matrix load, multiply, accumulate and store operations and allows for direct programming on Tensor Cores. cuBLAS and cuDNN libraries, on the other hand, shield users away from low-level CUDA programming and provides a very versatile, and highly-tuned, high-level user API for GEMM and other operations.

For a given GEMM in Eq. 4 with matrix size $M \times N$ for A, $N \times K$ for B, and $M \times K$ for C and D, if $M = N = K$, the total number of FLOPs performed in this kernel can be calculated as $M^3 \times 2$. This is an estimation without including the constant efficiency multiplications, which usually are performed on the CUDA Core, not Tensor Core, and are negligible. With the run time t , we can then estimate the FLOP/s performance of the kernel as $(M^3 \times 2)/t$ for a given matrix size in Fig. 2.

It is clear that as the matrix size increases, so does the performance of both *wmma* and *cuBLAS* approaches. At the largest with $M = N = K = 32768$, we have obtained 103.7 TFLOP/s at 96.5% of the theoretical peak from the *cuBLAS* approach, and 58 TFLOP/s at 54% from the *wmma* approach. This is largely due to the optimizations in *cuBLAS* such as the use of shared memory,

data padding (to avoid bank conflicts in shared memory), highly tuned thread block size, tile size, and other parameters.

For the rest of this paper, we will use 103.7 TFLOP/s as the Tensor Core peak; however, the 58 TFLOP/s performance provides an empirical upper bound for users who program in *wmma* on the Tensor Core.

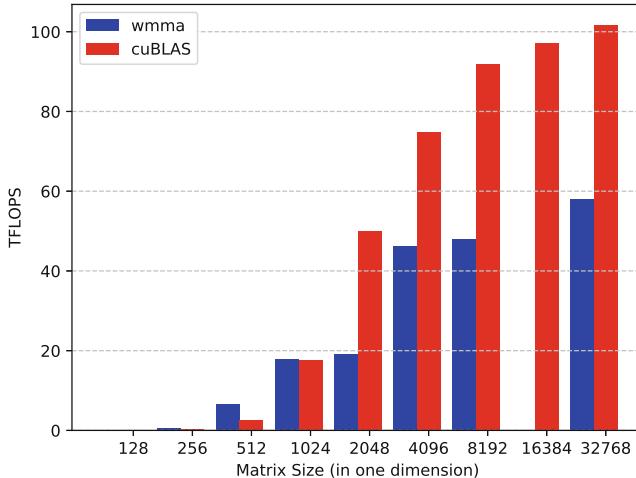


Fig. 2. Tensor Core performance as a function of matrix size for cuBLAS and hand-optimized WMMA implementations of matrix multiplication.

2.2 Nsight Compute Metrics for Application Characterization

The application characterization methodology for Roofline analysis on NVIDIA GPUs has been evolving with the developer toolchain change. The first proposed methodology was based on `nvprof` [13] in [39], and then an `Nsight Compute` [11] based methodology is developed at [14] and briefly presented in [36]. In this paper, we will discuss in detail how the `Nsight Compute` metrics can be used for hierarchical Roofline analysis on NVIDIA GPUs and demonstrate its effectiveness in analyzing deep learning applications.

The `Nsight` profiling toolkit is replacing `nvprof` as the new performance tool suite for NVIDIA GPU developers. It consists of three components, `Nsight Systems`, `Nsight Compute`, and `Nsight Graphics`, with the first two being most relevant to scientific application and machine learning application development. `Nsight Systems` can provide a system-wide visualization of application performance and help users identify issues such as insufficient parallelism on the GPU, unnecessary device-host data transfers, and inefficient kernel synchronization, while `Nsight Compute` dives a bit deeper and allows for the collection of more detailed performance metrics such as warp issues statistics, instruction pipeline utilization, and memory access pattern.

Between the two generations of developer tools, `nvprof` and `Nsight Compute` have a few major differences.

- `nvprof` uses CUPTI [10] while `Nsight Compute` is based on PerfWorks [12], a new framework for performance metric collection.
- The metrics in `Nsight Compute` are more nuanced than in `nvprof`, with some metrics broken down into more in `Nsight Compute`.
- The naming and organizing convention in `Nsight Compute` is more structured as well, with components such as unit, subunit, interface, counter name, rollup metric and submetric, used to distinguish different metrics.
- Kernel replay when multiple metrics are being collected, and profiling overhead, are more optimized in `Nsight Compute`, to provide faster and more accurate hardware and software counter measurements.

To construct a hierarchical Roofline on NVIDIA GPUs, we need to collect the following quantities: kernel run time, the total number of FLOPs performed in each kernel, and the number of bytes being read and written at each level of the memory hierarchy. With `Nsight Compute`, we can use this command to collect metrics listed in Table 2.

$$nv\text{-}nsight\text{-}cu\text{-}cli\text{-}metrics \textbf{metric} ./application \quad (5)$$

Kernel Run Time. As shown in Table 2, we use the metric `sm_cycles_elapsed.avg` to obtain the total number of elapsed `cycles` and its submetric `per_second` to get the `rate` (number of cycles per second), in order to calculate the kernel execution time:

$$time = cycles/rate \quad (6)$$

FLOPs. To count the number of FLOPs performed in the kernel, `Nsight Compute` doesn't provide a unified metric like `flop_count_dp` in `nvprof`. But for each floating-point precision (FP64, FP32 and FP16), it splits the measurement into three metrics based on the instruction type, addition, multiplication, and fused multiply-add (FMA). Note that each FMA is considered two FLOPs and the total number of FLOPs can be calculated as `add + 2 x fma + mul` for each data precision. Also, one can tell from the naming of the metrics that only non-predicated threads are counted in these FLOPs, i.e. masked operations are not included.

For Tensor Core, we count the number of warp instructions by using the `sm_inst_executed_pipe_tensor.sum` metric and the total Tensor Core FLOPs is

$$FLOP_{tc} = Inst_{tc} \times 512 \quad (7)$$

Table 2. Nsight compute metrics for hierarchical roofline

	Metrics
Time	sm_cycles_elapsed.avg sm_cycles_elapsed.avg.per_second
FP64 FLOPs	sm_sass_thread_inst_executed_op_hadd_pred_on.sum sm_sass_thread_inst_executed_op_hmul_pred_on.sum sm_sass_thread_inst_executed_op_hfma_pred_on.sum
FP32 FLOPs	sm_sass_thread_inst_executed_op_fadd_pred_on.sum sm_sass_thread_inst_executed_op_fmul_pred_on.sum sm_sass_thread_inst_executed_op_ffma_pred_on.sum
FP16 FLOPs	sm_sass_thread_inst_executed_op_hadd_pred_on.sum sm_sass_thread_inst_executed_op_hmul_pred_on.sum sm_sass_thread_inst_executed_op_hfma_pred_on.sum
Tensor Core FLOPs	sm_inst_executed_pipe_tensor.sum
L1 Cache	l1tex_t_bytes.sum
L2 Cache	lts_t_bytes.sum
HBM	dram_bytes.sum

Bytes. Metrics are listed in Table 2 for measuring the data movement on each level of the memory hierarchy.

For device memory (or HBM), L2 cache, and L1 cache, the latest **Nsight Compute** provides a unified byte metric for each of them to facilitate measurement. Note that shared memory transactions are not included in the current L1 metric.

Due to profiling overhead, it is recommended to restrict the number of kernels to run with **Nsight Compute** at a time, and these metrics can be collected on separate runs as well, as long as the execution of the application is deterministic. Also, note that as of 2020.1.0, **Nsight Compute** serializes multi-stream execution so certain performance gain due to kernel overlapping may be overlooked; however, the performance analysis in this paper is still insightful in understanding application performance on a kernel level.

3 Experimental Setup

3.1 Hardware and Software Configuration

Results presented in this paper are obtained from the Cori supercomputer, and in particular its GPU partition, at the National Energy Research Scientific Computing Center (NERSC), Lawrence Berkeley National Laboratory (LBNL). The GPU partition is primarily deployed for GPU porting, benchmarking, and testing efforts in the NERSC Exascale Science Application Program (NESAP). Each

node contains two Intel Xeon Gold 6148 Skylake CPUs, 384 GiB DDR4 memory, and 8 NVIDIA V100 GPUs. Each GPU has 16GiB of HBM2 memory and 80 SMs, and GPUs on a node are connected to each other in a ‘hybrid cube-mesh’ topology.

On the software side, we have used the TensorFlow 1 and PyTorch implementation of the climate image segmentation code in [4], and CUDA 10.2.89, cuDNN 7.6.5, Nsight Compute 2020.1.0, Python 3.7, PyTorch 1.5.0, and TensorFlow 1.15.0 for this study.

3.2 DeepCAM Benchmark

DeepCAM [4] is a deep learning benchmark extracted from the 2018 Gordon Bell winning project [27], used for detection, classification, and localization of extreme weather patterns in climate images. It has two different implementations, in TensorFlow and PyTorch, respectively with the PyTorch version being selected for MLPPerf [7] HPC benchmark suite. In this paper, we will compare the performance of these two implementations using the methodology presented in Sect. 2.2. To ensure a fair comparison, we have tuned the parameters to be as close as possible, for example, the number of layers in the encoder-decoder architecture, layer parameters, optimization algorithms, step rates, batch size, usage of batch norm, and Automatic Mixed Precision (AMP) settings.

The DeepCAM model is a deep neural network for semantic segmentation with an encoder-decoder architecture based on DeepLabv3+ [15]. The encoder is a ResNet-50 network with atrous spatial pyramid pooling. The decoder is a nine-layer network with convolutional and de-convolutional layers and two skip connections from the input and middle of the encoder.

To profile the code, the `profile-from-start` option is disabled in `Nsight Compute` and we use CuPy [32] to explicitly restrict the profiling region to include the iteration loop only. To have relatively stable run time behavior during profiling, we also set up a warm-up loop with five iterations before the target profiling loop. We collect only one metric during each execution to minimize the profiling overhead which will result in random algorithmic choices due to the TensorFlow runtime auto-tuning. To solve this issue, NVIDIA TensorFlow Determinism [5] is employed to get rid of this uncertainty.

If not otherwise stated, the default setting for the TensorFlow DeepCAM implementation is with AMP-enabled, and for PyTorch DeepCAM with AMP optimization level 01. The source code and full raw results are available at [4].

4 Results

In this section, we apply the `Nsight Compute` methodology in Sect. 2.2 on the DeepCAM benchmark and discuss its performance implications. Roofline ceilings are collected via ERT. On the following Roofline charts, each kernel is represented by a triplet of open circles (blue for L1, red for L2 and green for HBM), and the circle size is proportional to the kernel’s run time. Note that we

preset a minimum circle size to make all kernels visible on the plot, and that the real run time difference between large and small kernels can be more significant. Besides, there could be many invocations of the same kernel and the data presented on these Roofline charts is the aggregation of all these invocations of the same kernel. One should expect blue, red, and green circles near the L1, L2, and HBM ceilings respectively to show high memory utilization. Triplets of circles close to each other present a “streaming” data access pattern and indicate poor cache locality. Circles to the top right corner show superior performance over the others.

In the following subsections, we will discuss how performance is different in the forward and backward pass in both TensorFlow and PyTorch implementations, and the performance impact of the NVIDIA Automatic Mixed Precision package and the zero-AI kernels. Note that the backward pass for TensorFlow DeepCAM includes both gradient calculation and gradient update, whereas the PyTorch DeepCAM backward pass only includes gradient calculation (with its ‘optimizer’ being the gradient update step).

4.1 The TensorFlow Version of DeepCAM

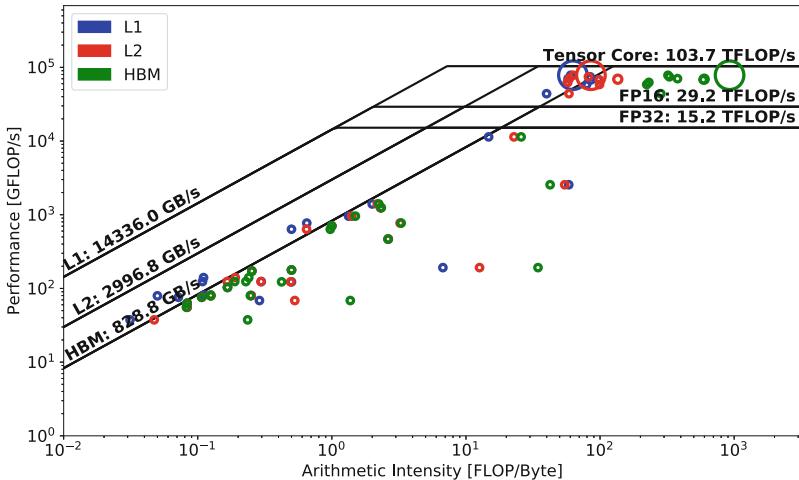


Fig. 3. Hierarchical Roofline of the TensorFlow DeepCAM in the forward pass with default configurations. The dominant kernel (with three largest circles) has very high Tensor Core utilization and consume 33% of the overall run time.

Figure 3 shows the hierarchical Roofline of the TensorFlow version of DeepCAM in its forward pass. The main computational kernel represented by the three large circles under the Tensor Core ceiling, indicates that it has very high Tensor Core utilization, whereas many of the other circles either do not use Tensor Core or

are bandwidth bound. This major kernel's L1 circle (in blue) slightly overlaps with its L2 circle (in red) indicating a relatively low L1 cache locality; however, the large gap between its L2 and HBM circles demonstrates that L2 cache misses rarely happened and that the kernel benefits from high L2 data locality. As for the rest of the kernels, their L1, L2, and HBM kernels are generally close to each other, implying a poor data locality across all levels of memory hierarchies ("streaming" operations).

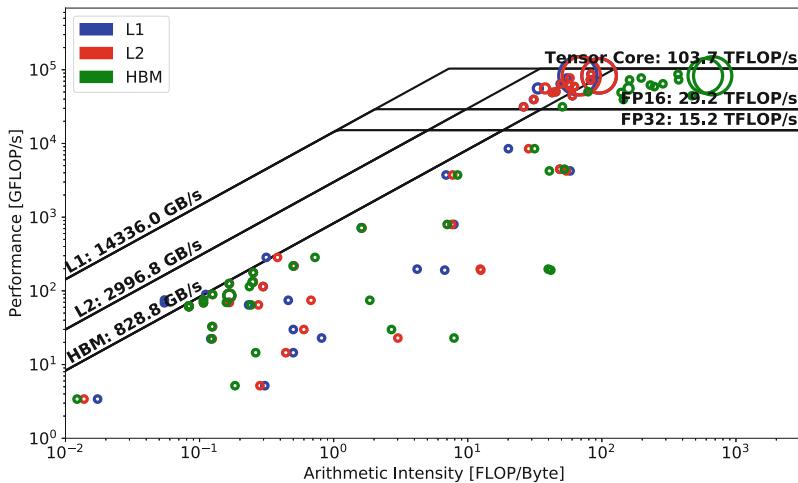


Fig. 4. Hierarchical Roofline of the TensorFlow DeepCAM in the backward pass with default configurations. There are more compute-intensive kernels than in the forward pass. Collectively they constitute 41.9% of the run time and attain near peak Tensor Core performance.

Figure 4 shows the corresponding backward pass of the TensorFlow DeepCAM. Instead of one single major kernel appearing in the forward pass, two very time-consuming kernels are found in the backward pass calculation. It is obvious that these two kernels both require longer run time than the major kernel in the forward pass (notice the size), which implies that the backward pass has more compute-intensive kernels than the forward pass and is generally more time-consuming. Compared to a few kernels using Tensor Core in the forward pass, we can find that more kernels benefit from the Tensor Core pipeline in the backward pass since they are sitting above the half-precision peak. Another observation is that more kernel invocations are involved in the backward pass than in the forward. Overall, we can conclude that in either forward or backward pass, the main computational kernels are compute-bound and are highly optimized for the underlying architecture.

4.2 The PyTorch Version of DeepCAM

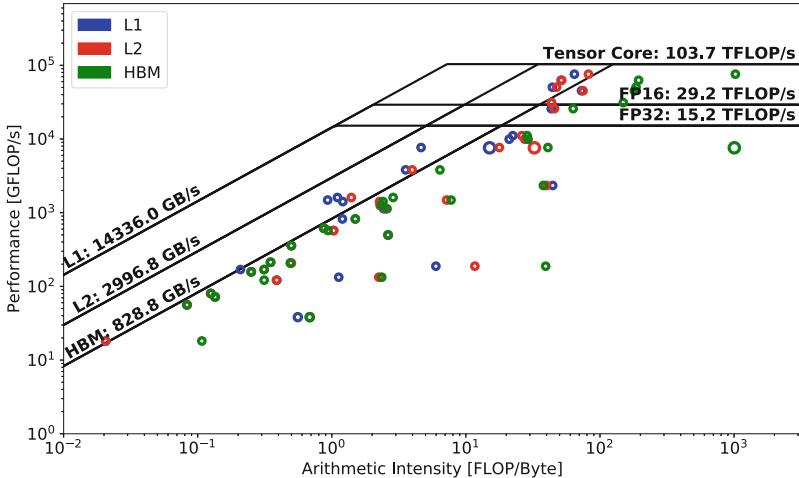


Fig. 5. Hierarchical Roofline of the PyTorch DeepCAM in the forward pass with default configurations. No single kernel requires significantly longer run time than the others (no extremely large circles).

Compared to the TensorFlow result (Fig. 3), no dominant kernels (kernel run time significantly larger than the others) can be found in the PyTorch forward pass (Fig. 5). The number one performance bottleneck (the largest circle triplet) is located slightly below the single-precision performance peak, and based on the proximity of symbols between different memory hierarchies, it has a better cache utilization than the dominant kernel in TensorFlow (even though it runs on the CUDA Core). Besides, similar to TensorFlow, a large number of trivial kernels are HBM-bound in the PyTorch implementation of DeepCAM.

Figure 6 shows the PyTorch DeepCAM performance in the backward pass, with default configurations. Surprisingly, the number one time-consuming kernel does not utilize Tensor Core and delivers only about 1 TFLOP/s performance. However, this implementation’s overall run time is still lower than that of the TensorFlow case, seen by the size of the circles, thanks to optimizations in other kernels or the overall execution of kernels.

Compared to TensorFlow, PyTorch has more flexibility when profiling the model, and the ‘optimizer’ step can be easily separated from the gradient calculation step in the back-propagation. The optimization step is mainly to update model parameters with newly calculated gradients and is usually low on arithmetic intensity. Figure 7 confirms this, where all the ‘optimizer’ kernels are memory-bound and have a much lower FLOP/s performance than some of the

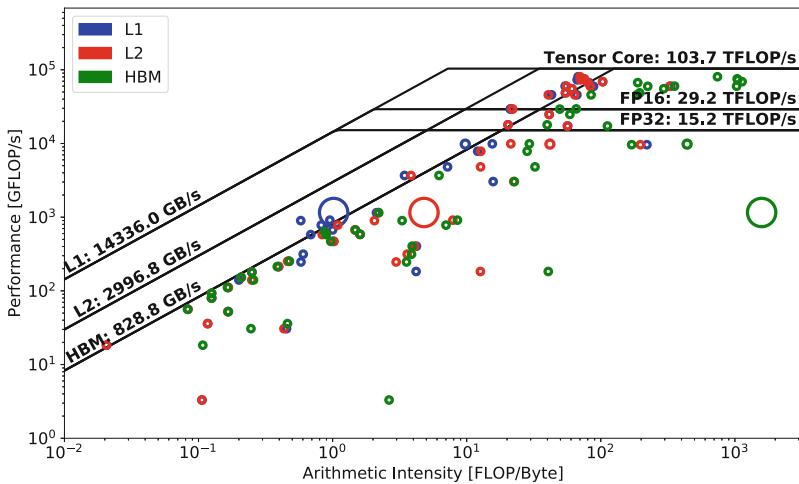


Fig. 6. Hierarchical Roofline of the PyTorch DeepCAM in its backward pass with default configurations. One can observe the highly compute intensive, but low performing kernel.

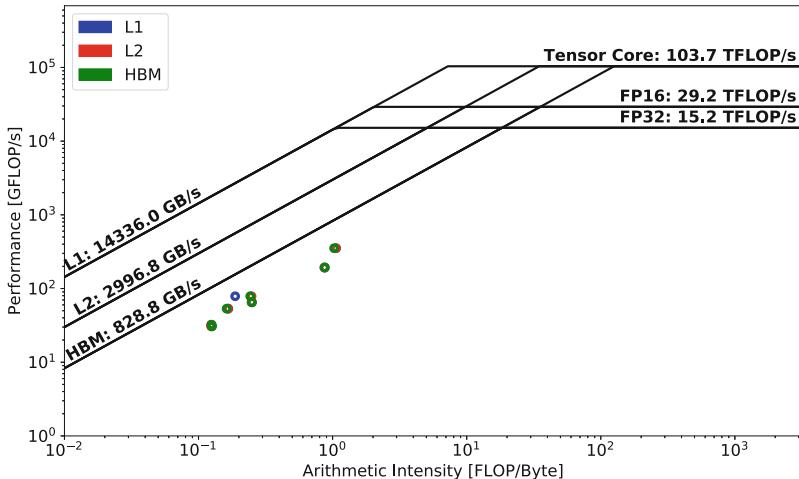


Fig. 7. Hierarchical Roofline of the PyTorch DeepCAM in its ‘optimizer’ step. The gradient update step consists of numerous streaming operations and has poor arithmetic intensity and FLOP/s performance.

kernels in Fig. 5 or Fig. 6. It should be noted that there are 2709 kernel invocations involved in this process, even though there are only a few circles visible. These kernel invocations have very similar arithmetic intensity and performance, and are thus overlapping.

4.3 Automatic Mixed Precision

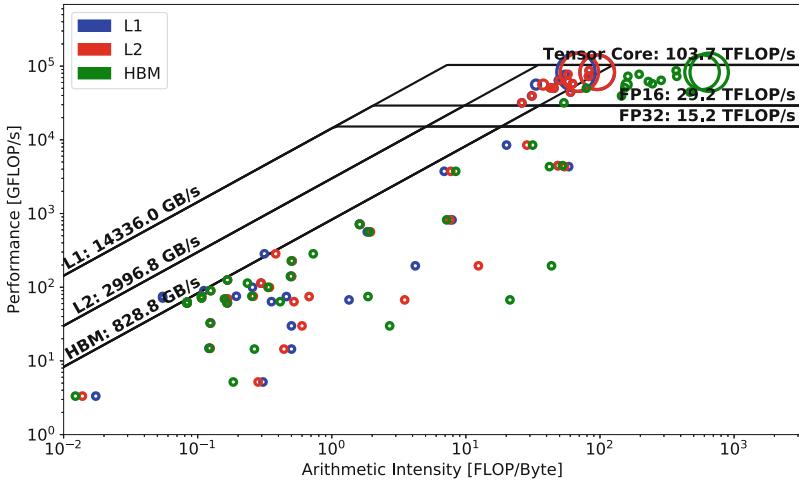


Fig. 8. Hierarchical Roofline of our FP16 implementation of DeepCAM in TensorFlow (backward pass). AMP (shown in Fig. 4) can deliver the same performance without manual type conversion.

The Automatic Mixed Precision (AMP) package developed at NVIDIA is dedicated to accelerating deep learning processes by partially converting single-precision data to half-precision to reduce data movement and improve computational throughput. It allows for automatic type conversion of certain model parameters and also implements schemes such as loss scaling to ensure numerical correctness and accuracy. We have implemented an FP16 version of DeepCAM in TensorFlow manually, by picking out the appropriate variables by hand and typecasting them explicitly. Figure 8 shows that the backward pass performance of this implementation is very close to that of the FP32 DeepCAM with AMP-enabled (shown in Fig. 4), demonstrating that even without the knowledge of the implementation details of the network, the AMP package can effectively apply type conversion and leverage lower-precision operations for performance.

AMP provides implementation for both TensorFlow and PyTorch, and for PyTorch, there are more detailed optimization levels, rather than just on or off. According to the AMP documentation [1], 00 level for PyTorch is used to establish a stable baseline for the auto mixed-precision acceleration; 01 follows a conservative type conversion and numerical properties are highly preserved; 02 however, implements a more aggressive FP32 to FP16 conversion and extra care needs to be taken for model convergence concerns.

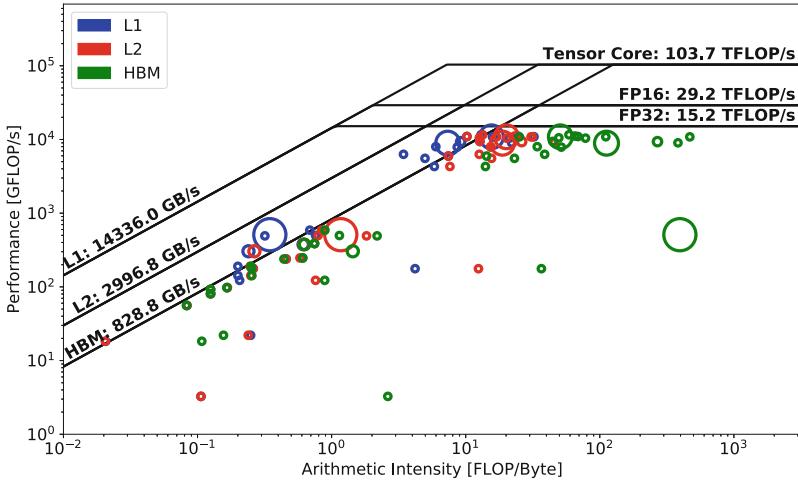


Fig. 9. Hierarchical Roofline of the PyTorch DeepCAM in its backward pass with AMP O0.

Our default setting is O1 and the backward pass performance of the PyTorch DeepCAM with this setting is shown in Fig. 6. From the O0 optimization level in Fig. 9, to the O1 in Fig. 6, kernel run time has been largely reduced and many kernels have been moved to execute on the Tensor Core, providing a much higher computational throughput and demonstrating the effectiveness of the O1 optimization level.

4.4 Zero-AI Kernels

Compared to traditional HPC applications where users usually have full control of kernel invocations, high-level Python-based deep learning frameworks tend to implicitly invoke many subsidiary kernels, either for data conversion or device-host transfer purposes. Table 3 shows the ratio of these kernel invocations to the total number of invocations. Around 40–50% of the invocations are for such zero-AI kernels, where no floating-point operation is performed. This may not inadvertently affect the overall performance much if these kernels are perfectly overlapped with other kernel executions, but it is very hard to achieve that in reality. As hardware constantly evolves, new computer architectures tend to provide higher and higher FLOP/s performance and bandwidth, but with less progressive improvement on kernel launch overhead. To avoid becoming overhead-bound, it is recommended that these deep learning applications avoid such “implicit” zero-AI kernels as much as possible by fusing them or overlapping with the non-zero-AI kernels.

Table 3. Zero-AI kernel Invocations in TensorFlow DeepCAM and PyTorch DeepCAM

TensorFlow DeepCAM	Forward	Backward ^a	Total	
zero-AI	304 (54.7%)	1833 (40.1%)	2137	
non zero-AI	252 (45.3%)	2740 (59.9%)	2992	
Total	556 (100%)	4573 (100%)	5129	
PyTorch DeepCAM	Forward	Backward	Optimizer	Total
zero-AI	437 (54.8%)	609 (38.7%)	0 (0%)	1046
non zero-AI	360 (45.2%)	966 (61.3%)	2709 (100%)	4035
Total	797 (100%)	1557 (100%)	2709 (100%)	5081

^aThis includes both gradient calculation and update, i.e. the backward pass and optimizer in the PyTorch case.

4.5 Overall Performance

Despite minor differences in implementation (even though we have tried to make an apples-to-apples comparison), the two codes, TensorFlow DeepCAM and PyTorch DeepCAM, have achieved similar runtime and convergence performance. The previous subsections presented a deep analysis of these two implementations on hierarchical Roofline, and it is discovered that TensorFlow tends to utilize Tensor Core more, compared to PyTorch, as seen by the locations of the most time-consuming kernels in Fig. 3, 4, 5 and 6. These two frameworks have similar cache utilization pattern on L1, L2 and HBM levels, with PyTorch having slightly more high-AI kernels scattered in the range of 100 FLOPs/Byte and 1000 FLOPs/Byte on Fig. 5 and Fig. 6.

Overall, similar numbers of kernels are launched in TensorFlow DeepCAM and PyTorch DeepCAM, with TensorFlow using over double the amount of zero-AI kernels than in PyTorch, 2137 versus 1046 in Table 3. These zero-AI kernels may have been launched over multiple streams and overlapped with computational kernels, however, reducing them could further improve the launch overhead and overall run time. These kernels are mostly used for converting data from one precision to another, or for rearranging data layout. They may be fused or done on the host (asynchronous to the GPU computation) in order to save run time.

Another note is that the NVIDIA AMP package has been proven to be very effective, through the comparison of Fig. 4 and Fig. 8 for TensorFlow, and Fig. 6 and Fig. 9 for PyTorch.

5 Conclusions

In this paper, we first revisited the need for mixed-precision performance analysis and extended ERT to incorporate single-precision, half-precision, and Tensor Core performance measurements. Then, based on the previous `nvprof` hierarchical Roofline methodology, we established a new `Nsight Compute` methodology

to collect Roofline data on NVIDIA GPUs. In the third part of this paper, we applied this new methodology to a representative real-life deep learning benchmark, DeepCAM, with its two implementations in TensorFlow and PyTorch. Results show that this new methodology is very effective in analyzing and better understanding the performance of deep learning applications. Useful performance insights are discussed, for example, computational characteristics of different stages of the training process, the performance impact of the automatic mixed precision (AMP) package and zero-AI kernels. This should be largely helpful to deep learning programmers and framework developers, as it captures data localities within each level of the cache hierarchy, demonstrates overall hardware utilization and indicates potential optimization efforts (get rid of zero-AI kernels to minimize kernel launch latency and improve overall FLOP rate).

In the future, we would like to extend the current `Nsight Compute` methodology to incorporate cross-node performance analysis. New methodologies for alternate architectures and mixed-precision performance ceilings in Roofline will be investigated as well.

Acknowledgment. This material is based upon work supported by the Advanced Scientific Computing Research Program in the U.S. Department of Energy, Office of Science, under Award Number DE-AC02-05CH11231. This research used resources of the National Energy Research Scientific Computing Center (NERSC) which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. We thank NVIDIA Corporation for their willingness to answer our myriad of questions on `Nsight` metrics.

References

1. apex.amp. Accessed 15 Oct 2020
2. CUDA C++ wmma API
3. CUDA cuBLAS Library
4. Deep Learning Climate Segmentation Benchmark
5. Deterministic Profiling for TensorFlow
6. Empirical Roofline Toolkit (ERT). Accessed 15 Oct 2020
7. MLPerf Benchmark
8. NERSC Roofline Model Documentation
9. Nsight compute cli - metric comparison. Accessed 15 Oct 2020
10. NVIDIA CUPTI API reference guide
11. Nvidia developer tools overview. Accessed 15 Oct 2020
12. PerfWorks measurement library for Nsight Compute
13. Profiler user's guide. Accessed 15 Oct 2020
14. Roofline Methodology on NVIDIA GPUs
15. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), September 2018
16. Chetlur, S., et al.: cuDNN: efficient primitives for deep learning. arXiv preprint [arXiv:1410.0759](https://arxiv.org/abs/1410.0759) (2014)

17. Choi, J.W., Bedard, D., Fowler, R., Vuduc, R.: A roofline model of energy. In: 2013 IEEE 27th International Symposium on Parallel and Distributed Processing, pp. 661–672 (2013)
18. Ben, M.D., Yang, C., Louie, S., Deslippe, J.: Accelerating large-scale GW calculations on hybrid GPU-CPU systems. *Bull. Am. Phys. Soc.* **65** (2020)
19. Ding, N., Williams, S.: An instruction roofline model for GPUs. In: 2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS), pp. 7–18. IEEE (2019)
20. Doerfler, D., et al.: Applying the roofline performance model to the Intel Xeon Phi knights landing processor. In: International Conference on High Performance Computing, pp. 339–353. Springer (2016)
21. Gayatri, R., Yang, C., Kurth, T., Deslippe, J.: A case study for performance portability using OpenMP 4.5. In: International Workshop on Accelerator Programming Using Directives, pp. 75–95. Springer (2018)
22. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
23. Ibrahim, K.Z., Williams, S., Oliker, L.: Performance analysis FF GPU programming models using the roofline scaling trajectories. In: International Symposium on Benchmarking, Measuring and Optimization, pp. 3–19. Springer (2019)
24. Javed, M.H., Ibrahim, K.Z., Lu, X.: Performance analysis of deep learning workloads using roofline trajectories. *CCF Trans. High Perform. Comput.* **1**(3), 224–239 (2019)
25. Joubert, W., et al.: Attacking the opioid epidemic: determining the epistatic and pleiotropic genetic architectures for chronic pain and opioid addiction. In: SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 717–730. IEEE (2018)
26. Koskela, T., et al.: A novel multi-level integrated roofline model approach for performance characterization. In: International Conference on High Performance Computing, pp. 226–245. Springer (2018)
27. Kurth, T., et al.: Exascale deep learning for climate analytics. In: SC18: International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 649–660. IEEE (2018)
28. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. *Handb. Brain Theory Neural Netw.* **3361**(10), 1995 (1995)
29. Lopes, A., Pratas, F., Sousa, L., Ilic, A.: Exploring GPU performance, power and energy-efficiency bounds with cache-aware roofline modeling. In: 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 259–268 (2017)
30. Madsen, J.R., et al.: Timemory: modular performance analysis for HPC. In: International Conference on High Performance Computing, pp. 434–452. Springer (2020)
31. Tesla NVIDIA. V100 GPU architecture. The world's most advanced data center GPU. version WP-08608-001_v1. 1. NVIDIA. Aug, p. 108 (2017)
32. Okuta, R., Unno, Y., Nishino, D., Hido, S., Loomis, C.: CuPy: a numpy-compatible library for NVIDIA GPU calculations. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS) (2017)
33. Wang, Y., Yang, C., Farrel, S., Zhang, Kurth, Y.T., Williams, S.: Time-based roofline for deep learning performance analysis. In: 2020 IEEE/ACM Deep Learning on Supercomputers Workshop (2020, Submitted)

34. Williams, S., Waterman, A., Patterson, D.: Roofline: an insightful visual performance model for floating-point programs and multicore architectures. Technical report, Lawrence Berkeley National Lab. (LBNL), Berkeley, CA, USA (2009)
35. Yang, C.: 8 Steps to 3.7 TFLOP/s on NVIDIA V100 GPU: Roofline analysis and other tricks
36. Yang, C.: Hierarchical roofline analysis: how to collect data using performance tools on Intel CPUs and NVIDIA GPUs
37. Yang, C., Friesen, B., Kurth, T., Cook, B., Williams, S.: Toward automated application profiling on cray systems. In: Cray User Group Conference (CUG) (2018)
38. Yang, C., et al.: An empirical roofline methodology for quantitatively assessing performance portability. In: 2018 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC), pp. 14–23. IEEE (2018)
39. Yang, C., Kurth, T., Williams, S.: Hierarchical roofline analysis for GPUs: accelerating performance optimization for the NERSC-9 perlmutter system. *Concurr. Comput. Pract. Exp.* **32**, e5547 (2019)



Data Augmentation for Short-Term Time Series Prediction with Deep Learning

Anibal Flores^(✉), Hugo Tito-Chura, and Honorio Apaza-Alanoca

Universidad Nacional de Moquegua, Moquegua, Peru

Abstract. In this paper, a hybrid data augmentation technique for short-term time series prediction is proposed in order to overcome the underfitting problem in deep learning models based on recurrent neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The proposal hybrid technique consists of the combination of two basic data augmentation techniques that are generally used for time series classification, these are: time-warping and jittering. Time-warping allows the generation of synthetic data between each pair of values in the time series, extending its length, while jittering allows the synthetic data generated to be non-linear. To evaluate the proposal technique, it's experimented with three non-seasonal short-term time series of Perú: CO₂ emissions per capita, renewable energy consumption and Covid-19 positive cases, it is considered that predicting non-seasonal time series is more difficult than seasonal ones. The results show that the regression models based on recurrent neural networks using the selected time series with data augmentation improve results between 16.318% and 42.1426% .

Keywords: Data augmentation · Time-warping · Jittering · Deep learning · Time series prediction

1 Introduction

Short-term time series are present in many areas of scientific endeavor, this because in many cases gathering large amounts of data takes months and years; this results in the fact that it is not possible to successfully apply artificial intelligence techniques based on Deep Learning, since training models with little data leads to the problems known as overfitting [1] and underfitting. Overfitting occurs when the model manages to learn the training data well but predicts the test data poorly, that is, there were data that the model could not learn because they were not present in the training data. And underfitting when the model does not learn the training cases due to insufficient data, and therefore poorly predicts the test data. In short-term time series prediction with deep learning, the problem of underfitting usually occurs.

In image classification models, the most common problem presented is overfitting, which was solved through data augmentation techniques and these consisted of image-resizing, image-rotation, left-right flipping, and zooming [2] among others. Something similar happens in time series classification, here the data augmentation techniques have

been classified as basic and advanced. The basic techniques are based on the domain of time and frequency and some examples of them are presented in works such as: [1, 3–5], these include time-warping, window-warping, scaling, rotation, jittering, permutation, etc. The advanced techniques are based on time series decomposition or machine learning methods as shown in [6] and [7].

Regarding the problem of underfitting in short-term time series prediction, no data augmentation works have been found, despite the fact that in [8] the authors refer to papers [9] and [10] as proposals. Thus, in the present work, inspired by some data augmentation techniques for time series classification, a hybrid technique based on time-warping and jittering is generated. Using only time-warping would allow increasing the number of items in the time series, but in a somewhat linear way, which may not help to achieve satisfactory results. However, combining time-warping and jittering techniques would help make synthetic data more robust [3]. Using only Jittering would not allow increasing the number of items in the time series, so it would not contribute to the objective of the study.

The proposal technique allows generating two types of synthetic data: linear synthetic data and random synthetic data. Linear synthetic data is generated trying to preserve the original characteristics of the time series such as the trend, unlike the Synthetic Minority Over-Sampling Technique (SMOTE) [11] for classification models, which locates the synthetic data randomly on the line that joins two records, the proposal inserts them in such a way that the distance between each linear value is equal or similar. The random synthetic data is used to complete the gaps generated after the insertion of the linear values in the part time series, these random data are values between the prior and the next value to avoid a high dispersion in the synthetic data. Figure 1 shows the CO₂ emissions time series with different amounts of augmented items (5, 10 and 15).

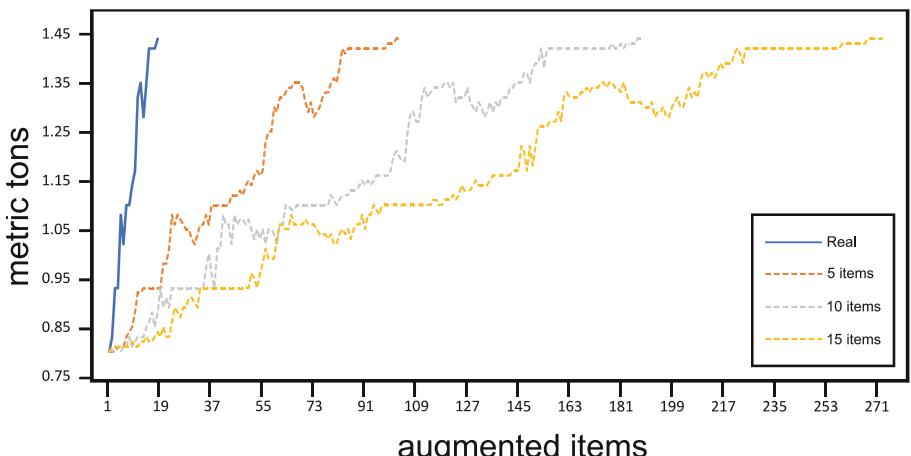


Fig. 1. CO₂ emissions time series with different amounts of augmented items. A time series of 18 iItems with 5 augmented items will reach a length of 103, with 10 augmented items 188 and with 15 augmented items 273.

This work is limited to the analysis and evaluation of the proposal technique with short-term time series, for this, three time series have been selected. An exhaustive comparison of the results with other techniques for the prediction of this type of time series is not carried out.

Regarding the structure of the work, this has been organized in different sections: Sect. 2, corresponds to the Related Work, there the works related to the data augmentation proposal are briefly described. Section 3 describes the proposal technique in detail. In Sect. 4, two models based on recurrent neural networks LSTM and GRU are implemented to predict time series with and without data augmentation. In Sect. 5, the results achieved are analyzed and described. Finally, the conclusion of the study and the future work that can be carried out from the proposal is shown.

2 Related Work

Some works related to the area of data augmentation for time series classification are described below.

In [1] the authors propose a method of generating synthetic time series to overcome the problem of overfitting in time series classification, it consists of the fusion of parts from different time series. Because the technique does not allow increasing the number of items in time series, it was not considered as the basis for the proposal technique of this paper.

In [5] the authors propose a data augmentation technique called window-warping for time series classification which consists of selecting a time series segment and stretching or shorten it. This technique is inspired by time-warping, but because it can alter the frequency of the real data, like the previous work it was discarded.

In [7] the authors propose a data augmentation technique, which consists of pattern mixing, creating synthetic time series from the average of two random sub-optimally aligned patterns. As this technique did not allow to increase the time series length, it was discarded as the basis for the proposal technique of this work.

In [3] the authors propose a time series classification framework based on time series augmentation, the techniques for time series augmentation consist of jittering, scaling, rotation and time-warping; The time series correspond to people walking, the objective is improving accuracy of personal identification with deep learning. From this work, time-warping and jittering was selected as inspiration for the implementation of the proposal presented in this paper.

Some works related to short-term time series prediction are described below:

In [12] the authors propose a new method to forecast short rainfall time series. The proposal is based on Bayesian methods and works with datasets at least 36 samples. The results show that the Bayesian proposal technique outperforms the techniques with which it was compared such as Artificial Neural Networks (ANN), ARMA and others.

In [13], the authors propose the use of the recurrent neural network LSTM combined with Average True Range (ATR) index. The data used correspond to three load energy time series of three periods of 2, 3 and 3 months, respectively. The results achieved show that the proposal LSTM+ ATR surpasses the other techniques with which it was compared such as ARIMA, SARIMA, Feed Forward Neural Network and others.

In [14], the authors fundamentally compare two techniques: Nearest Neighbor Regression (NNR) and Autoregressive Integrated Moving Average (ARIMA), these models are tested using different hyperparameters, e.g., the number of lags, the size of the training data set. The results show that NNR is better than ARIMA for solar irradiance time series prediction.

3 Background

3.1 Data Augmentation

Data augmentation is the artificial generation of data through disturbances in the original data. This allows the training data set to be increased both in size and diversity. In computer vision, this technique became a standard for regularization, and also to improve performance and combat overfitting on CNNs.

For time series classification the basic data augmentation techniques, they include time-warping, window-warping, scaling, rotation, jittering, permutation, etc. some of them can be seen in Fig. 2.

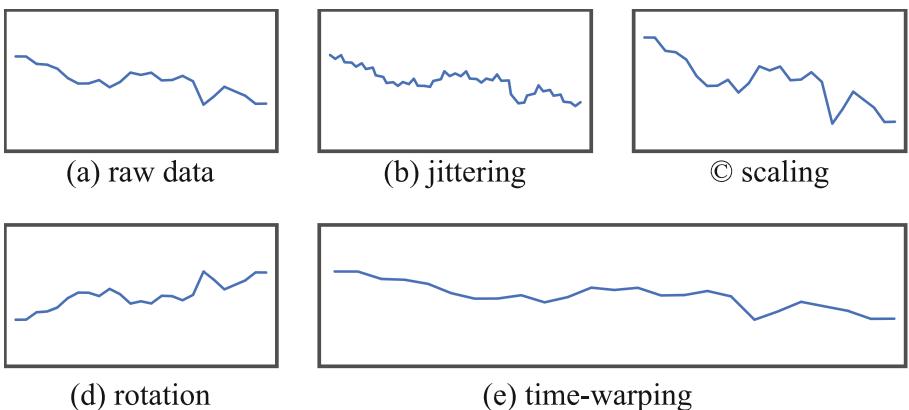


Fig. 2. Basic data augmentation techniques for classification of time series

From this set of basic data augmentation techniques, time-warping and jittering were chosen for the proposal technique for time series prediction in such a way that the underfitting problem can be overcome.

3.2 Deep Learning

Deep learning [15, 16] is a set of machine learning algorithms that attempts to model high-level abstractions in data using computational architectures that support multiple and iterative nonlinear transformations of data expressed in matrix or tensor form.

Deep learning models use a cascade of layers with non-linear processing units to extract and transform variables. Each layer uses the output of the previous layer as

input. Algorithms can use supervised learning or unsupervised learning, and applications include data modeling and pattern recognition.

The main deep learning algorithms are classified into deep neural networks, convolutional neural networks and recurrent neural networks. In this work it is experimented with the results of the proposal data augmentation technique in recurrent neural networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU).

4 The Proposal

The data augmentation proposal technique for short-term time series is described in detail below:

4.1 Parameters

The parameters of the technique are: time series, synthetic block size, the sub-block size and precision.

- a) Time series are the original values with an insufficient number of items to adequately train a recurrent neural network.
- b) The synthetic block size indicates the number of synthetic items that will be inserted between each pair of elements in the original time series.
- c) The sub-block size indicates the maximum size of every sub-block, and depending on the value of the block size, the number of synthetic elements may be less than the value set for this parameter, in Fig. 3 it can be seen, sub-block size is defined as 6, but according synthetic block size is estimated as 5.
- d) The precision is used to establish the number of decimal places to be considered in the generation of synthetic items.

4.2 Generating Linear Synthetic Values

Once the parameters of the technique have been established, the synthetic values are generated according to the value established for the synthetic block size parameter. The first synthetic values that were generated are the linear ones and they are distributed in each block according to the estimated sub-block size. It can be seen in Fig. 2 in the linear synthetic values section.

4.3 Generating Non-linear Synthetic Values

Once the linear synthetic values have been generated, the non-linear synthetic values are generated to complete the gaps that are seen in the linear synthetic values section in Fig. 2, these are estimated from random numbers between the prior and next value of each gap, in such a way that a high dispersion of the data is avoided.

This process is repeated for each pair of items in the original or real time series.

Complete Javascript source code for proposal technique can be appreciated in Appendix A.

Original time series



Parameters for data augmentation

synthetic block size=17 sub-block size=6

synthetic block size=17



Linear synthetic values



*sub-block size for
synthetic block size=17*

Random non-linear synthetic values

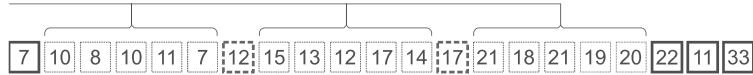


Fig. 3. Parameters of proposal technique with linear and random non-linear synthetic items

5 Experimentation

The activities carried out for the experimentation stage are briefly described below.

5.1 Times Series Selection

The selected time series correspond to data from Peru and these are: CO₂ emissions per capita¹ (55 items), renewable energy consumption² (25 items) and COVID-19 positive cases (30 items). The first two were obtained from the World Bank repository and the third from the repository of the Health Ministry of Peru³. Table 1 shows time series data.

5.2 Test and Training Data

The data for the training and testing phases is structured according to Table 2.

¹ <https://data.worldbank.org/indicator/EN.ATM.CO2E.PC?locations=PE>.

² <https://data.worldbank.org/indicator/EG.FEC.RNEW.ZS?locations=PE>.

³ <https://www.datosabiertos.gob.pe/dataset/casos-positivos-por-covid-19-ministerio-de-salud-minsa>.

Table 1. Selected time series data

Time series	Range	Data
CO ₂ emissions per capita	[1960–2014]	0.80 0.83 0.93 0.93 1.08 1.02 1.10 1.10 1.14 1.17 1.32 1.35 1.28 1.35 1.42 1.42 1.42 1.44 1.35 1.30 1.37 1.34 1.28 1.08 1.07 0.99 1.08 1.25 1.19 1.01 0.95 0.90 0.89 1.01 0.98 0.98 0.99 1.09 1.08 1.13 1.15 1.01 1.00 0.96 1.15 1.33 1.24 1.52 1.44 1.80 1.98 1.70 1.87 1.92 2.05
Renewable energy consumption	[1990–2014]	39.43 39.41 37.31 37.08 35.93 33.28 31.71 31.74 32.69 30.68 32.16 34.84 34.17 34.82 32.61 32.73 33.88 32.37 25.72 28.06 30.8 29.55 28.25 25.98 26.0
Covid-19 positive cases	[2020/03/11–2020/04/10]	5 5 16 15 28 15 31 28 89 29 55 45 32 21 64 100 55 36 181 98 115 258 91 181 151 535 280 393 1388 914

Table 2. Data for testing and training

Time series	Training items	Testing items
CO ₂ emissions per capita	27	28
Renewable energy consumption	12	13
Covid-19 positive cases	15	15

5.3 Data Augmentation

At this stage, the data augmentation proposal is worked with block size of 15 and 30 items for the three chosen time series. Likewise, six items were considered as a maximum sub-block size for all cases. The results of this stage for each time series are:

03 time series for training, 01 without data augmentation, 01 with 15 augmented items and 01 with 30 augmented items.

03 time series for testing, 01 without data augmentation, 01 with 15 augmented items and 01 with 30 augmented items.

5.4 Prediction with Deep Learning

The deep learning models that were chosen for this stage are two basic recurrent neural network models, such as the Long Short-Term Memory (LSTM) of two layers, and the Gated Recurrent Unit (GRU) also of two layers, they were implemented in Google Colab and the source code is similar to that implemented in works like [16, 17] and others, they can be seen in Fig. 4 and Fig. 5.

5.5 Evaluation of Results

In the case of time series with synthetic data (15 and 30 items), the predictions corresponding to the synthetic data are removed and the models are only evaluated with non-synthetic data.

```

model=Sequential()
model.add(LSTM(units=30,return_sequences=True,input_shape=(features_set.shape[1],1)))
model.add(Dropout(0.2))

model.add(LSTM(units=30))
model.add(Dropout(0.2))

model.add(Dense(units=1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.fit(features_set, labels, epochs = 100, batch_size = 30)

```

Fig. 4. Google Colab Python code for two-layer LSTM Model

```

model=Sequential()
model.add(GRU(units=30,return_sequences=True,input_shape=(features_set.shape[1],1)))
model.add(Dropout(0.2))

model.add(GRU(units=30))
model.add(Dropout(0.2))

model.add(Dense(units=1))

model.compile(optimizer = 'adam', loss = 'mean_squared_error')
model.fit(features_set, labels, epochs = 100, batch_size = 30)

```

Fig. 5. Google Colab Python code for two-layer GRU Model

Two metrics are used to evaluate results: Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) that are estimated through Eqs. (1) and (2).

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n-1} (Pi - Ri)^2}{n}} \quad (1)$$

$$MAPE = \left[\frac{1}{n} \sum_{i=0}^{n-1} \frac{|Ri - Pi|}{|Ri|} \right] * 100 \quad (2)$$

Where:

n: number of predicted values.

Pi: vector of predicted values.

Ri: vector of original values.

6 Results

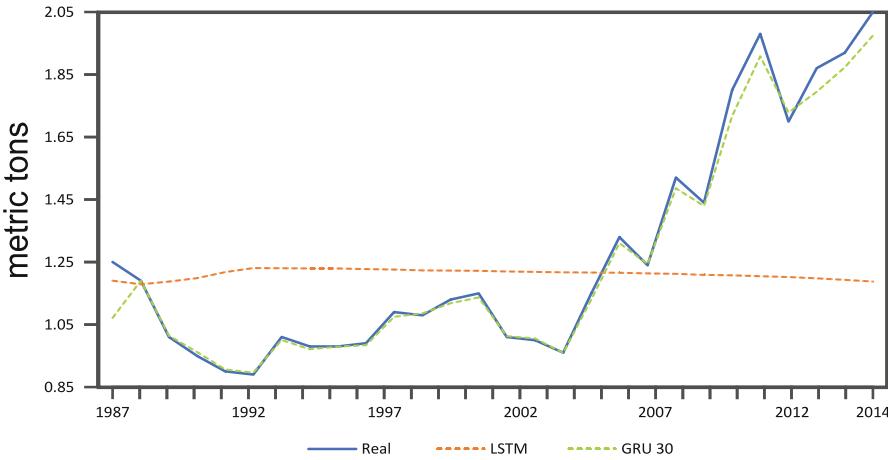
The results achieved for the experimented time series in the previous section are described and analyzed below.

According Table 3, it can be seen that for the CO₂ emissions per capita time series, for 15 augmented items the LSTM model improves by 18.5477%; while GRU model improves 19.2955%. For 30 augmented items, LSTM model improves 19.4169% and GRU model improves 19.7151%. The best prediction model for this time series is GRU with 30 augmented items, the Root Mean Square Error (RMSE) is 0.0468, and the

Table 3. Predictions for the experimented time series

Prediction model	Augmented items					
	0		15		30	
	RMSC	MAPE	RMSE	MAPE	RMSE	MAPE
<i>CO₂ Emissions per capita</i>						
LSTM	0.3705	21.4071	0.0627	2.8594	0.0516	1.9902
GRU	0.3760	21.5498	0.0534	2.2543	0.0468	1.8347
<i>Renewable energy consumption</i>						
LSTM	6.4552	19.8697	1.3347	3.5517	0.7395	2.0878
GRU	6.5437	20.1808	0.5290	1.4455	0.4146	1.1076
<i>COVID-19 positive cases</i>						
LSTM	472.9441	65.7041	291.9623	27.4225	314.4513	27.6599
GRU	476.7713	71.1219	304.2882	29.8284	307.9650	28.9793

Mean Absolute Percentage Error (MAPE) is 1.8347%. Likewise, it can be observed that without data augmentation, the LSTM predictions are better than those of GRU that clearly show underfitting. In general, for CO₂ emissions time series, the deep learning models with data augmentation: LSTM 15, LSTM 30, GRU 15 and GRU 30 overcame the underfitting problem, resulting GRU 30 model as the best of them, it can be seen in Fig. 6.

**Fig. 6.** Comparison of predicted values for CO₂ emissions per Capita in Peru.

In the case of the renewable energy consumption time series, comparing deep learning models with underfitting with respect to models without underfitting, for 15 augmented item, LSTM model improves 16.318% and GRU model improves 18.7353%; while for 30 augmented items, LSTM model improves 17.7819% and GRU model improves 19.0732%. Similarly, to the CO₂ emissions time series, the best prediction model for this time series is GRU with 30 augmented items, the Root Mean Square Error (RMSE) is

0.4146, and the Mean Absolute Percentage Error (MAPE) is 1.1076%. The results show that the proposal technique enables deep learning models to overcome the underfitting problem. Figure 7 shows a graphical comparison of the prediction results in this time series.

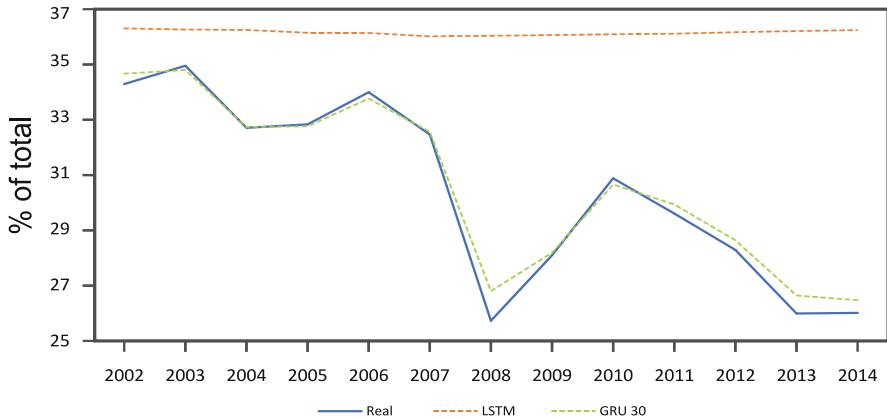


Fig. 7. Comparison of predicted values for renewable energy consumption in Peru.

Finally, in the case of the third time series, the one corresponding to Covid-19 positive cases, for 15 augmented items, LSTM model improves 38.2816% and GRU model improves 41.2935%; while for 30 augmented items, LSTM model improves 30.0442% and GRU model improves 42.1426%. The best prediction model is LSTM with 15 augmented items the Root Mean Square Error (RMSE) is 291.9623, and the Mean Absolute Percentage Error (MAPE) is 27.4225. In a similar way to the two previous time series, the data augmentation allowed to overcome the problem of underfitting, so the deep learning models obtained satisfactory results. Figure 8 graphically shows the prediction results in this time series.

Figure 6, Fig. 7 and Fig. 8 clearly show how the predictions of the models based on recurrent neural networks without data augmentation have the underfitting problem, they cannot satisfactorily predict the testing data for each of the time series that were experimented. However, the time series with data augmentation of 15 and 30 synthetic items allows to considerably increase the precision of the LSTM and GRU models. GRU performed best for the first two time series, but LSTM was best for the third time series (COVID-19).

The time series that were best predicted are the first two, although the improvement in the percentage of error was not as high as in the third. The third time series obtained the highest percentage increase in predictions, with LSTM 15 it reached an improvement of 38.2816% and with GRU 15 and improvement of 41.2935%. Thus, in future works, the number of items for the training phase could be increased and the precision of the random synthetic values could be improved in order to get better results.

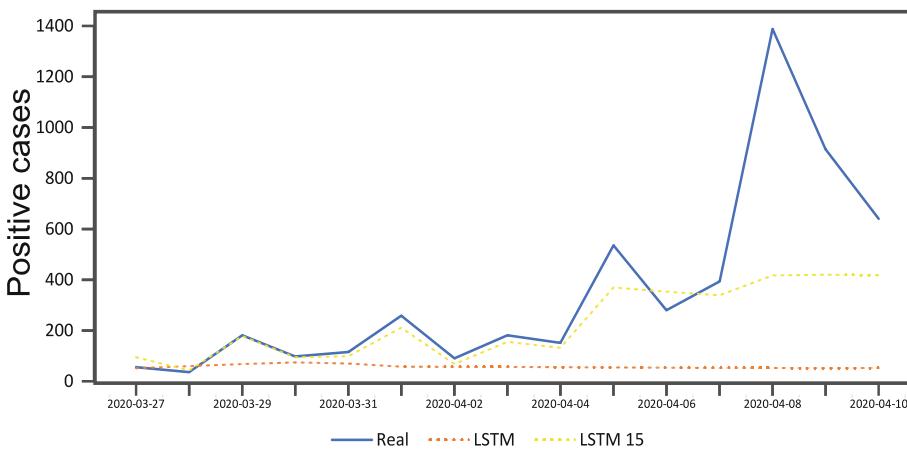


Fig. 8. Comparison of predicted values for Covid-19 positive cases in Peru.

7 Conclusion

After observing the results achieved, it is concluded that the proposal data augmentation technique for univariate time series prediction allows to overcome the underfitting problem for the three time series analyzed in this study. The data augmentation allows to improve the precision of models based on recurrent neural networks (LSTM and GRU) enormously, going from being unsatisfactory models for short-term time series prediction to excellent models for this type of tasks.

8 Future Work

One of the main weaknesses of the proposal technique is the case of repeating items, the proposal technique uses each pair of values of a time series, and when finding repeated items, it will generate similar values for all block-size, this can lead to bias in the training stage and in model predictions. For future work, a simple solution to this problem could be performed, it can go backwards until it finds a different element, or else it could go forward until it finds a different value, in such a way that it never works with repeated values.

Also, it would be important to analyze the precision of recurrent neural networks with higher amounts of data augmentation, for example, 50, 100, 200, etc.

Appendix

APPENDIX 1. Javascript Code for Proposal Technique of Data Augmentation

```

//public variables
var prcsn;
var ts;
var block_size;
var s_block_size=6;
var brks=0;
var posbrks;
var valbrks;

//initialization of variables
function init()
{
    posbrks=new Array();
    valbrks=new Array();
    block_size=parseInt(document.getElementById('size').value);
    prcsn=parseInt(document.getElementById('prcsn').value)
    getBrkPositions();
    yts=document.getElementById("yts").value;
    ts=yts.split(" ");
    vda=generate_da();
    document.getElementById("results").innerHTML=vda
}

//positions of linear values
function getBrkPositions()
{
    brks=Math.floor(block_size/(s_block_size+1));
    batch=Math.floor((block_size-brks)/(brks+1));
    resbatch=Math.floor((block_size-brks)%(brks+1));
    pos=batch;
    resbrks=Math.floor(block_size%(s_block_size+1));
    for(i=0;i<brks;i++)
    {
        if(resbrks==0)
        {
            posbrks[i]=pos;
            pos+=batch+1;
        }
        else
        {
            add=0;
            if(resbatch>0)
            {
                resbatch--;
                add=1;
            }
            posbrks[i]=pos+add;
            pos+=batch+add+1;
        }
    }
}

#Generating synthetic values
function generate_da()
{
    da_vector=new Array();
    prior=parseFloat(ts[0]);
    next=parseFloat(ts[1]);
    for(i=1;i<ts.length;i++)
    {
        v=fillGap(prior,next);
        nv=v.length;
        if(i<(ts.length-1))

```

```

        v.splice(nv-1,1);
        v3=da_vector.concat(v);
        da_vector=v3;
        prior=next;
        next=parseFloat(ts[i+1]);
    }
    return da_vector;
}

#generating linear and non-linear values
function fillGap(prior,next)
{
    b=0;
    factor=(next-prior)/(brks+1);
    factor=parseFloat(factor.toFixed(prcsn));
    priorval=prior;
    for(a=0;a<brks;a++)
    {
        valbrks[a]=priorval+factor;
        priorval+=factor;
    }
    vector=new Array();
    vector.push(prior);
    ppos=-1;
    npos=posbrks[0];
    siz=npos-(ppos+1);
    priorval=prior;
    nextval=valbrks[0];
    for(z=1;z<=(brks+1);z++)
    {
        for(y=1;y<=siz;y++)
        {
            num=getRnd(priorval,nextval);
            vector.push(num);
        }
        vector.push(nextval);
        priorval=nextval;
        if(npos<posbrks[posbrks.length-1])
        {
            nextval=valbrks[z];
            ppos=npos;
            npos=posbrks[z];
            siz=npos-(ppos+1);
        }
        else
        {
            ppos=npos;
            npos=block_size;
            nextval=next;
            siz=npos-(ppos+1);
        }
    }
    return vector;
}

#random values generator
function getRnd(prior,next)
{
    p=1;
    switch(prcsn)
    {
        case 1:
            p=10; break;
        case 2:

```

```

        p=100; break;
    case 3:
        p=1000; break;
    case 4:
        p=10000; break;
    }
    prior=prior*p;
    next=next*p;
    min=prior;
    max=next;
    if(prior>next)
    {
        min=next;
        max=prior;
    }
    num=Math.floor(Math.random()*(max-min+1)+min);
    num=(num/p).toFixed(prcsn);
    return parseFloat(num);
}

```

References

1. Yeomans, J., Thwaites, S., Robertson, W.S.P., Booth, D., Ng, B., Thewlis, D.: Simulating time-series data for improved deep neural network performance. *IEEE Access* **7**, 131248–131255 (2019)
2. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**(60), 1–48 (2019)
3. Rashid, K.M., Louis, J.: Times-series data augmentation and deep learning for construction equipment activity recognition. *Adv. Eng. Inform.* **42**, 1–12 (2019)
4. Iwana, B.K.; Uchida, S.: Time series data augmentation for neural networks by time warping with a discriminative teacher. arxiv.org (2020)
5. Rashid, K.M., Louis J.: Window-warping: a time series data augmentation of IMU data for construction equipment activity identification. In: 36 International Symposium on Automation and Robotics in Construction (ISARC 2019), Banff, Canada (2019)
6. Le Guennec, A., Malinowski, S., Tavenard, R.: Data augmentation for time series classification using convolutional neural networks. In: ECML/PKDD Workshop on AALTD (2016)
7. Kamycky, K., Kapuscinski, T., Oszust, M.: Data augmentation with suboptimal warping for time-series classification. *Sensors* **20**(98), 1–15 (2020)
8. Wen, Q., Sun, L., Song, X., Gao, J., Wang, X., Xu, H.: Time series data augmentation for deep learning: a survey. Arxiv.org, pp. 1–7 (2020)
9. Salinas, D., Flunkert, V., Gasthaus, J.: DeepAR: probabilistic forecasting with autoregressive recurrent networks. Arxiv.org, pp. 1–12 (2017)
10. Wen, R., Torkkola, K., Narayanaswamy, B., Madeka, D.: A multi-horizon quantile recurrent forecaster. Arxiv.org, pp. 1–9 (2018)
11. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)
12. Rodriguez, C., Tupac, Y., Pucheta, J., Juarez, G., Franco, L., Otaño, P.: Time-series prediction with BEMCA approach application to short rainfall series. In: IEEE LA-CCI, Arequipa, Peru (2017)
13. Panapongpakom, T., Banjerpongchai, D.: Short-term load forecast for energy management system using time series analysis and neural network method with average true range. In: ICA-SYMP, Bangkok, Thailand (2019)

14. Hans, C., Klages, E.: Very short term time-series forecasting of solar irradiance without exogenous inputs. Arxiv.org, pp. 1–13 (2020)
15. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**, 1798–1828 (2013)
16. Flores, A., Tito, H., Centyy, D.: Comparison of hybrid recurrent neural networks for univariate time series forecasting. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) IntelliSys 2020. AISC, vol. 1250, pp. 375–387. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-55180-3_28
17. Flores, A., Tito, H., Centyy, D.: Recurrent neural networks for meteorological time series imputation. *Int. J. Adv. Comput. Sci. Appl. (IJACSA)* **11**(3), 482–487 (2020)



On the Use of a Sequential Deep Learning Scheme for Financial Fraud Detection

Georgios Zioviris¹(✉), Kostas Kolomvatsos², and George Stamoulis¹

¹ Department of Electrical and Computer Engineering, University of Thessaly,
Glavani 37, 38221 Volos, Greece
{gzioviris,georges}@uth.gr

² Department of Informatics and Telecommunications, University of Thessaly,
Papasiopoulou 2-4, 35131 Lamia, Greece
kostasks@uth.gr

Abstract. Forecasting fraud detection has never been more essential for the finance industry than today. The detection of fraud has been a major concern for the banking industry due to the high impact on banks' revenues and reputation. Fraud can be related with an augmented financial risk, which is often underestimated until it is too late. Recently, deep learning models have been introduced to detect and forecast possible fraud transactions with increased efficiency compared to the conventional machine learning methods and statistics. Such methods gain significant popularity due to their ability to estimate the unknown distribution of the collected data, thus, increasing their capability of detecting more complex fraud events. In this paper, we introduce a novel multistage deep learning model that combines a feature selection process upon an Autoencoder model and a deep convolutional neural network to detect frauds. To manage highly unbalanced datasets, we rely on the Synthetic Minority Over-sampling Technique (SMOTE) to oversample our dataset and adjust the class distribution delivering an efficient classification approach. We describe the problem under consideration and our contribution that provides a solution for it. An extensive set of experimental scenarios are adopted to reveal the performance of the proposed scheme exposing the relevant numerical results. A comparative assessment is used for proving the superiority of our model compared with a Support Vector Machine (SVM) scheme, a classical CNN model and the results of two researches that use the same dataset.

Keywords: Fraud detection · Deep learning · Autoencoder · Convolutional neural network · Dimensionality reduction

1 Introduction

Basel III has implemented a more strict regulatory framework, under which, every financial organization is obliged to comply, in order to improve the banking sector's ability to absorb shocks arising from financial and economic stress,

whatever the source, thus, reducing the risk of spillover from the financial sector to the real economy [8]. One of the main danger in the banking sector is fraud. Despite the serious risk that a fraud presents to business, many organisations still do not have formal systems and procedures in place to prevent, detect and respond to fraud [4]. Until now, internal rating models based on the standard linear econometric approach have been generally shown to exhibit poor performance in forecasting loss given default [3]. Credit fraud has become a significant factor to the overall bank risk, mostly because of the recent severe financial crisis, which revealed a significant amount of fraud cases. Subsequently, the avoidance of credit fraud has become a critical factor in the banking industry, thus, the development of an accurate assessment model to classify any transaction as normal or fraudulent has become a priority for various institutions. Deep learning has been used in many domains, e.g., image processing, speech recognition, and Natural Language Processing (NLP), with many proposed deep learning schemes like autoencoders [9], Deep Convolutional Neural Networks (CNNs) [34], Recurrent Neural Networks (RNNs) [29].

The relevant models see increased popularity and usefulness, largely as the result of more powerful computers, larger datasets and techniques to train deeper networks [20]. Deep networks perform much better with more data than classical machine learning algorithms. More often than not, the performance of a deep network model can be increased by feeding more training data, while with ‘typical’ ML algorithms, this approach is not efficient enough requiring more complex methods to improve accuracy.

This paper concentrates on fraud detection using a combination of deep learning algorithms in order to improve the classification accuracy and most importantly, eliminate Type I and Type II errors, which represent the number of fraud transactions that are classified as regular and the opposite [6]. We propose a novel deep learning model, which combines a deep autoencoder that is responsible for the feature reduction of the dataset and a CNN, that classifies the transactions as frauds or not. We rely on a model to deal with the complexity of the underlying data, i.e., the combination of multiple parameters before we are in a position to efficiently detect a fraud. Our deep learning approach tries to, initially, detect features that are significant as depicted by the incoming dataset, then, we rely on them to perform the final classification. Our feature selection approach is adopted to reduce the computational burden of the upcoming classification process performed by our CNN. We strategically decide to rely on the most significant features in the monitoring process at the edge. This way, our monitoring mechanism can focus only on a subset of data instead of the entire dataset. Due to highly unbalanced datasets, we first perform the Synthetic Minority Over-sampling Technique (SMOTE) [2] to balance the dataset, by oversampling the minority class. Hence, any decision is realized upon the ‘complex’ trends of data that may dynamically change over time. To the best of our knowledge, this is the first time that a feature selection upon an autoencoder’s outcomes is combined with a classification task based on a CNN for fraud detection. Compared with other efforts in the respective literature, we go a step forward and identify the ‘hidden’ aspects of a fraud detection event.

For instance, we argue for a more efficient dimensionality reduction technique via the deep autoencoder that could lead to more accurate classification results compared to legacy ML schemes. The following list reports on the contributions of this paper:

- We provide an autoencoder for performing dimensionality reduction and identify the most significant features in the streaming datasets. The autoencoder efficiently learns the representation of data under consideration and generates the reduced encoding that is used to reconstruct the original input.
- We support the desired classification process with a CNN that detects fraud events. The proposed CNN adopts a connectivity pattern of the involved neurons that can learn specific parts of data and, finally, detect potential frauds. Additionally, the aforementioned connectivity patterns incorporate overlaps to learn the connections between features. Eventually, we are able to ‘cover’ all the aspects of data under consideration over which we try to take the final decision.
- We provide an extensive evaluation process that exposes the pros and cons of our scheme. We provide a set of experimental scenarios accompanied by the corresponding numerical outcomes.

The rest of the paper is organized as follows. Section 2 reports on the related work, while Sect. 3 presents the problem that we are trying to resolve. Section 4 describes the proposed algorithm, while in Sect. 5, we present our experimental results. Finally, in Sect. 6, we present our conclusions by giving insights of our future research plans.

2 Related Work

Many techniques have been applied to maximize the detection rate of frauds through the adoption of ML and deep learning techniques. The interested reader can find a relevant review in [5]. ML models involve Neural Networks (NNs), decision trees, genetic algorithms, while outlier detection techniques can be also the basis for the identification of frauds [5]. There has been a surge in recent years in the use of Machine Learning (ML) tools for evaluating credit risk and detecting fraud. [4]. Studies of credit risk show that while ML models outperform traditional ones, their performance depends on the specific ML scheme, the environment, and the sample used in the analysis [10, 12, 14]. ML techniques have been proven to perform better than traditional statistical techniques, both in classification and also in predictive accuracy [30]. Currently, deep learning forms a state of the art technology as an ‘extension’ of the ‘legacy’ ML models that give better performance in multiple research fields. Additionally, in [11], the authors present an experimental comparison of classification algorithms such as random forests and gradient boosting classifiers for unbalanced scoring datasets. The presented outcome depicts that random forests and gradient boosting algorithms outperform the remaining algorithms involved in the comparison (e.g., C4.5, quadratic discriminant and k-nearest neighbours - kNNs). In [7], the authors conclude

that Support Vector Machines (SVMs) improve the accuracy of events detection compared to logistic regression, linear discrimination analysis and kNNs. A survey on SVMs introduces the application of the technology and the techniques adopted to predict defaults using broad and narrow definitions [13]. Another effort, presented in [21], tries to evaluate ML models (SVMs, bagging, boosting, and random forests) to predict bankruptcy one year prior to the event, and compare their performance with results from discriminant analysis, logistic regression, and NNs. The aforementioned attempt evaluates the strength of ensemble models over single classifiers focusing on the accuracy of the outcomes. In [18], the authors present a comprehensive review of financial fraud detection research using such data mining methods, with a particular focus on computational intelligence (CI)-based techniques. In [15], the authors proposed the PrecisonRank and the total detection cost as the correct metrics for measuring the detection performance in credit datasets, while in [23], the authors perform an effective learning strategy for addressing the verification latency and the alert-feedback interaction problem. In [31] the authors expand the labeled data through the social relations to get the unlabeled data and propose a semi-supervised attentive graph neural network, named SemiGNN to utilize the multi-view labeled and unlabeled data for fraud detection. Moreover, they propose a hierarchical attention mechanism to better correlate different neighbors and different views. The authors of [27] use various machine learning algorithms, with and without the usage of the AdaBoost & majority voting algorithm, in order to detect fraudulent transactions.

In recent years, deep learning models have been also evaluated for fraud detection. CNNs appear for the first time in [1]. Their target is to manage every data observation as a ‘picture’ in the two-dimensional space. CNNs have been applied to almost every task possible, such as image recognition, object detection, classification, etc. A denoising autoencoder for credit risk analysis has been introduced to remove the noise from the dataset [24]. Denoising autoencoders often yield better representations when trained on corrupted versions of a dataset, thus, they can capture information and filter out noise more effectively than traditional methods [24]. A deep autoencoder and a Restricted Boltzmann Machine (RBM) that can reconstruct normal transactions to finally find anomalies, have been applied to a credit card dataset for fraud detection [25]. The authors conclude that the deep autoencoder and the RBM outperform other techniques if the available datasets are large enough to train them efficiently [25]. Sparse autoencoders and Generative Adversarial Networks (GANs) have been also adopted to detect potential frauds [28]. The discussed model can achieve higher performance than other state-of-the-art one-class methods such as one class Gaussian Process (GP) and Support Vector Data Description (SVDD). In [26], the authors introduce a hybrid ‘Relief-CNN’ model, i.e., a combination of two (2) techniques: a CNN and the Relief algorithm. The utilization of the relief algorithm can efficiently reduce the size of an image pixel matrix, which can reduce the computational burden of the CNN.

3 Problem Definition

Before we describe the technical aspect of our model, let us describe what a fraud is. A fraud event is committed to a non legitimate transaction involving a credit or a debit card. The credit/debit card can be authorised, where the genuine customer processes a payment to another account which is controlled by a criminal or unauthorised person. Obviously, the account holder does not provide authorisation for the payment to proceed and the transaction is carried out by a third party. Financial institutions record millions of transactions per day, thus, they should be supported by advanced methodologies to detect fraud on the fly.

In this paper, we are trying to implement a model that performs better than the conventional ML methods while being faster than the ‘time consuming’ deep learning methods. We take advantage from a feature selection scheme performed by a deep autoencoder. The training process of a typical CNN requires a lot of time; in many cases it requires days, of even weeks in order to be complete.

The monitoring mechanism observes huge volumes of data generated by electronic transactions. Data are not only characterized by an increased number of ‘tuples’ but also by an increased number of features in each ‘tuple’. Actually, we consider a multivariate scenario upon data $\mathbf{x} = [x_1, x_2, x_3, \dots, x_M]$ where M is the number of features and $x_i \in \mathbb{R}$. Our target is to detect a transaction \mathbf{x}° which is recognized as a fraud upon the ‘experience’ of our model. In other words, fraud is a special case of outliers detection, i.e., \mathbf{x}' significantly deviates from the ‘statistics’ of past transactions. The high imbalanced distribution of the classification classes (i.e., fraud or normal transaction) generates new challenges that may be met adopting a feature selection methodology.

The goal of the proposed approach is to detect the most appropriate (or stable) features that are not dependent on the dataset. However, to avoid to select features in favor of the dominant class (as fraud events are far less than the normal transactions), we rely on the SMOTE technique to train our model. Data are fed in the proposed autoencoder to perform the most efficient feature selection, then, $\mathbf{x}' = [x_1, x_2, x_3, \dots, x_L], L \leq M$ is classified through the adoption of an CNN.

4 The Proposed Methodology

4.1 The Training Process

Our approach is a sequential streaming data management model upon a multi-stage *SMOTE - Autoencoder - CNN* scheme as illustrated by Fig. 1. The proposed model combines three (3) different stages to, finally, conclude to the final classification outcome.

Obviously, a preprosessing step is necessary to prepare our data before they are the subject of the envisioned processing activities. All features are normalized into the unity interval.

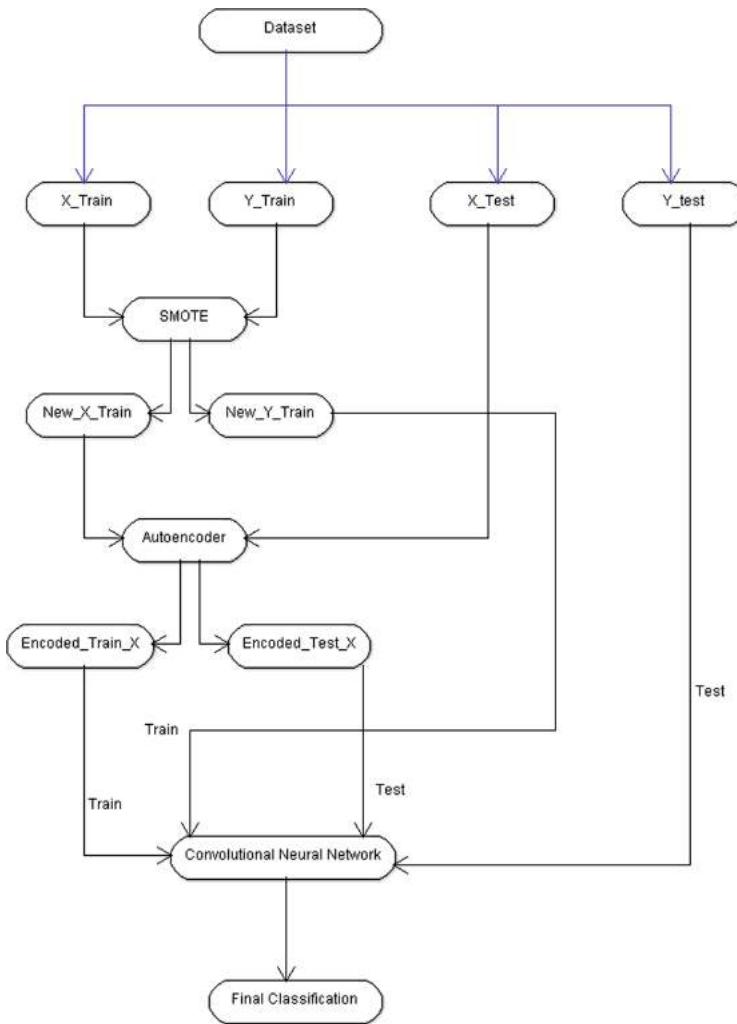


Fig. 1. The architecture of our model.

In the first step, and to tackle the unbalanced dataset problem, we create new instances of the minority class, while simultaneously, we reduce the instances of the majority class by adopting the SMOTE technique [2]. Then, in the second step, data are processed by the autoencoder, in order to perform a dimensionality reduction, thus, to make our system capable of managing huge volumes of streaming data. After the completion of the training process of the autoencoder, we perform the necessary transformations, in order to input the dataset into the CNN model. The transformed output of the hidden layer of the autoencoder is then transferred into the CNN in order to get the final decision, i.e., the classification of a transaction to a fraud event or not.

In the particular dataset, 30 features are included, while we conclude with only 15 of them, reducing the data space upon which we deliver the final classification outcome. To the best of our knowledge, it is the first time that the aforementioned models are combined to create a powerful mechanism for fraud detection. The discussed combination is performed with and without the use of SMOTE in the training process to adjust the class distribution of our dataset.

4.2 Dataset Oversampling

An unbalanced dataset could be a common problem met in ML and deep learning applications. Training a ML algorithm with such a dataset often results in a particular bias towards the majority class. To tackle the problem of an imbalanced dataset, SMOTE was introduced in [2]. The technique is based on a kNN model using the Euclidean distance between data points in the feature space. It is one of the most popular over sampling techniques adopted to overcome the problem of unbalanced datasets in ML schemes. For every minority instance, i.e., an observation that belongs to the minority class, k of nearest neighbours are detected, such that they belong to the same class where the minority class is over sampled. Over sampling is performed by taking each minority sample and introducing synthetic instances along the line segments joining any/all of the k minority class nearest neighbours [2]. Depending upon the required number of over samples, instances from the k nearest neighbours are randomly chosen.

4.3 The Proposed Autoencoder for Feature Selection

Autoencoders have always been part of the NN family and, traditionally, they are used for dimensionality reduction. Low dimensionality representations can improve performance on many tasks such as classification while consuming less memory and runtime. An autoencoder is a NN trained to attempt to copy its input to its output. Internally, it involves a hidden layer \mathbf{h} that describes a ‘code’ used to represent the input. The network may be viewed as the scheme containing two parts: an encoder function $\mathbf{h} = \mathbf{f}(\mathbf{x})$ and a decoder that produces a reconstruction of the initial input $\mathbf{r} = \mathbf{g}(\mathbf{h})$.

The following equations hold true:

$$\epsilon : X \rightarrow F \quad (1)$$

$$\tau : F \rightarrow X \quad (2)$$

$$\epsilon, \tau = \underset{\alpha, \lambda}{\operatorname{argmin}} \|X - (\tau \circ \epsilon)X\|^2 \quad (3)$$

The encoder function, denoted by ϵ , maps the original data X to a latent space F which is present at the hidden layer that acts as a ‘bottleneck’ of the data before they are reproduced by the decoder. The decoder function, denoted by τ , maps the latent space F at the bottleneck to the output. The output, in this case, is the same as the input. Hence, we are basically trying to recreate the

original data points after some generalized non-linear compression performed by the hidden neurons of our autoencoder. The encoding network can be represented by a standard NN function transferred through an activation function, where z is the latent dimension, i.e., $z = \sigma(Wx + b)$. Similarly, the decoding network can be represented in the same fashion, but with different weights, biases, and potentially activation functions. The decoding phase can be represented by the following equation: $x' = \sigma'(W'z + b')$.

The loss function can then be written in terms of the aforementioned NN functions, and it is the loss function that we use to train the NN through the standard back-propagation process. Using back-propagation, our algorithm continuously trains itself by setting the target output values to be equal the defined inputs. This forces the hidden encoding layer to use dimensionality reduction to eliminate noise and reconstruct the inputs. The following equations holds true:

$$L(x, x') = \|x - x'\|^2 = \|x - \sigma'(W'(\sigma(Wx + b)) + b')\|^2 \quad (4)$$

In our autoencoder, we use four layers (30, 27, 24, 21, nodes each) before we reach the final hidden layer that has only 15 nodes. For the implementation of our autoencoder, we adopt the Exponential Linear Unit (ELU) activation function, as it performs better than other options such as ReLU, Sigmoid and Tanh in our experiments. ELU is a function that tends to converge the cost to zero faster than others and produce more accurate results. Different than other activation functions, ELUs have negative values, which allows the network to push the mean activation closer to zero. Therefore ELUs decrease the gap between the normal gradient and the unit natural gradient and, thereby speed up learning [19].

4.4 The Proposed CNN

A CNN is a class of deep NNs, most typically applied in image and video recognition, recommender systems, image classification, medical image analysis, NLP and financial time series. The CNN employs a mathematical operation called convolution i.e., a specialized type of linear operation instead of general matrix multiplication, in at least one of its layers. A CNN consists of an input, an output and multiple hidden layers containing a series of convolutional layers that convolve with a multiplication or other dot product calculations. The activation function is often a Rectified Linear Unit (ReLU), and is subsequently followed by additional convolutions like pooling layers, fully connected layers and normalization layers. Mathematically, CNNs involve a sliding dot product or cross-correlation. This has significance for the indices within the matrix, in this, it affects how weights are set at a selected index point [20].

In general, convolution is an operation, that is described by the latter expression, which is best described as a smoothed estimate of our input function $x(t)$,

$$s(t) = \int x(a)w(t-a)da \quad (5)$$

where $w(a)$ is the Kernel function in the form of a valid probability density function, while $s(t)$ is the output. The convolutional operation, more often than not, is denoted by an asterisk:

$$s(t) = (x * w)(t) \quad (6)$$

It is more realistic though, to work on discrete values, as more datasets cannot provide measurements for every instance. So, alternatively, we can define the discrete convolution as:

$$s(t) = (x * w)(t) = \sum x(a)w(t - a) \quad (7)$$

Usually, we use convolutions for multiple axis at a specific time. So, the above functions can be expressed as:

$$S(i, j) = (I * K)(i, j) = \sum \sum I(m, n)K(i - m, j - n) \quad (8)$$

Or equivalently, exploiting the commutative ability of the convolution operation:

$$S(i, j) = (K * I)(i, j) = \sum \sum I(i - m, j - n)K(m, n) \quad (9)$$

The motivation behind the use of such a technique, is that convolution leverages three important ideas that can help improve a ML system, i.e., (i) sparse interactions; (ii) parameter sharing; and (iii) equivariant representations [20].

In our model, the output of the previous step (i.e., the output of the autoencoder) is fed into our CNN having an input layer, two (2) hidden layers and an output layer. In each layer, we implement a pooling layer, batch normalization and the dropout method to avoid overfitting. In the first two (2) layers, we adopt the ReLU activation function and for the output layer the activation process is performed by a Sigmoid function. Our decision for adopting the specific activation functions is concluded through an extensive experimentation that reveals their performance for the specific problem. The model loss is calculated with the binary cross entropy loss function (cross-entropy minimization is frequently used in optimization and rare event probability estimation).

5 Experimentation and Evaluation

5.1 Experimental Setup and Performance Metrics

We report on the evaluation of the proposed model upon a real dataset. Our dataset contains credit card transactions made in September 2013 by European cardholders collected during a research collaboration of Worldline and the Machine Learning Group (<http://mlg.ulb.ac.be>) of Université Libre de Bruxelles on big data mining and fraud detection [15–17, 22, 23, 32, 33]. This dataset depicts transactions occurred in two (2) days, where 492 frauds out of 284,807 transactions are present. The dataset is highly unbalanced, the positive class (frauds)

account for 0.172% of all the available transactions. It contains only numerical variables fed into our encoder. The feature ‘Class’ is the response/classification variable taking values equal to 1 in case of a fraud and 0, otherwise. In our experimentation, we perform feature normalization in the available features applying min-max normalization (i.e., we subtract the minimum of every feature, then dividing by the range of the feature).

6 performance metrics are adopted to evaluate our model, i.e., precision (ϵ), recall (ζ), F1-score (δ), accuracy (α), MCC (μ) & AUC’s score. Accuracy is the fraction of predictions our model got right. Precision is the fraction of true frauds amongst all samples which are classified as frauds, while recall is the fraction of frauds, which have been classified correctly over the total amount of frauds. F1-score is a measure that combines precision and recall. The Matthews Correlation Coefficient (MCC) is a machine learning measure which is used to check the balance of the binary (two-class) classifiers. It takes into account all the true and false values that is why it is generally regarded as a balanced measure which can be used even if there are different classes. AUC provides an aggregate measure of performance across all possible classification thresholds. The area under the curve (often referred to as simply the AUC) is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming ‘positive’ ranks higher than ‘negative’). It is also common to calculate the Area Under the ROC Convex as any point on the line segment between two prediction results that can be achieved by randomly using one or the other system with probabilities proportional to the relative length of the opposite component of the segment. The following equations hold true:

$$\alpha = \frac{TP + TN}{TP + TN + FP + FN} \quad (10)$$

$$\epsilon = \frac{TP}{TP + FP} \quad (11)$$

$$\zeta = \frac{TP}{TP + FN} \quad (12)$$

$$\mu = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}} \quad (13)$$

$$\delta = 2 \cdot \frac{\epsilon \cdot \zeta}{\epsilon + \zeta} \quad (14)$$

In the above equations, TP (True Positive) is the number of frauds which have been classified correctly. FP (False Positive) is the number of normal/valid transactions which have been classified as frauds. FN (False Negatives) is the number of frauds which have been classified as normal ones. TN (True Negatives) is the number of normal transactions that have been classified as normal.

5.2 Performance Assessment

Our set of experiments is performed with & without the use of the SMOTE technique. In Table 1, we present the classification results of our models, using 6 metrics for comparison reasons. We can see that our model with SMOTE (*SMOTE - AE - CNN*), performs better in terms of ζ , while *CNN* is better in terms of ϵ & δ . The *AE - CNN* performs better in terms of α & μ .

Table 1. Comparison between our models and the classical SVM & CNN Model

Models	FN	TP	TN	FP	ζ	ϵ	δ	α	μ	AUC
AE-CNN	28	111	85.287	17	79.86%	86.72%	83.15%	99.96%	85.92%	96.20%
SMOTE-AE-CNN	18	117	84.936	372	86.67%	23.92%	37.49%	99.54%	45.40%	98.01%
SVM	31	106	85.289	17	77.37%	86.17%	81.53%	99.94%	81.63%	97.94%
CNN	35	112	85.287	9	76.19%	92.56%	83.58%	99.95%	83.95%	98.00%

Figure 2 and 3 show the performance of both stages of our model. As shown in Fig. 2, the performance loss of the autoencoder is extremely low, while the overall model loss is shown in Fig. 2 (right). In addition, as shown in Fig. 3, we got an area under the curve (AUC) of 96.20%. The results of the confusion matrix are deployed in Table 1 and Fig. 3.

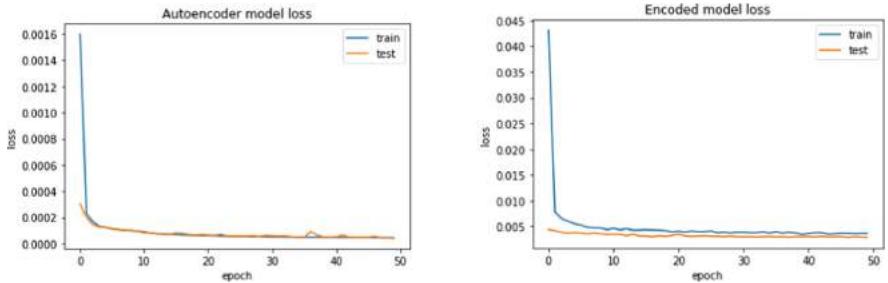


Fig. 2. Our model's losses (left: autoencoder - right: autoencoder & CNN)

Figure 4 and 5 show the performance of both stages of our model with the SMOTE technique. As shown in Fig. 4, the performance loss of the autoencoder is, again, extremely low, while the overall model loss is shown in Fig. 4 (right). In addition, as shown in Fig. 5, we got a AUC of 98,01%. The results of the confusion matrix are deployed in Table 1 and in Fig. 5 (right).

5.3 Comparative Assessment

Subsequently, our models are compared with the results that the authors of [25, 27] present in their research work, considering that the same dataset is used,

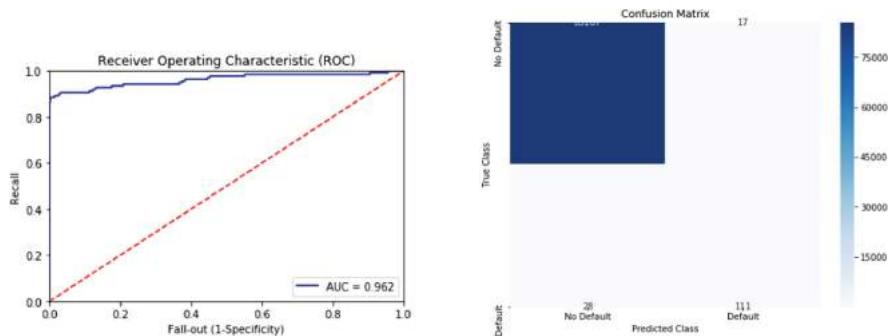


Fig. 3. Results for our autoencoder - CNN model (left: ROC curve loss - right: confusion matrix)

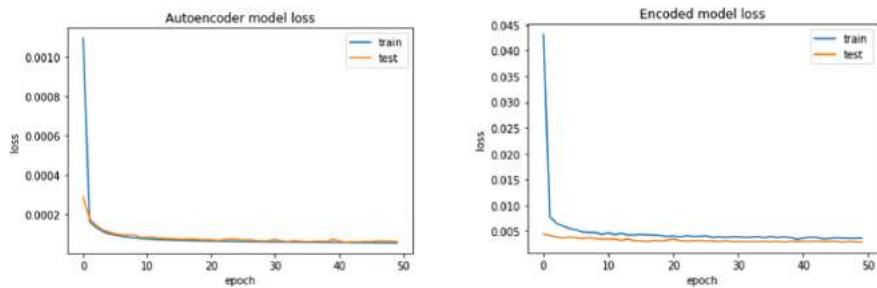


Fig. 4. Results when adopting SMOTE (left: autoencoder - right: autoencoder - CNN)

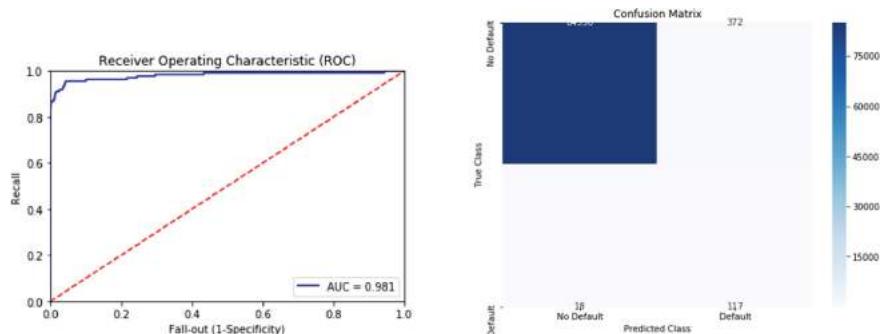


Fig. 5. Experimental evaluation when adopting SMOTE (left: ROC curve loss - right: confusion matrix)

so it easy to compare their results with ours. In [25] the authors use a deep autoencoder and a RBM model in order to outperform other techniques, while the only metric that is provided is the AUC score. In [27], the authors use various machine learning algorithms, with and without the usage of the AdaBoost & majority voting algorithm, in order to detect fraudulent transactions. The metrics that are provided are the Matthews Correlation Coefficient (MCC) and accuracy. None of those papers provide sufficient information about precision, recall and the F1 score. Nevertheless, in our work we try to compare our results, even though there is a lot of information that is omitted, in order to make a thorough and complete comparison. The results of the comparison are presented in Tables 2, 3, 4 and 5.

Table 2. Comparison between our models' results and the results from [25], i.e. AE & RBM

Models	AUC
AE-CNN	96.20%
SMOTE-AE-CNN	98.01%
AE [25]	96.03%
RBM [25]	95.05%

Comparison between our models' results and the results from [27].

Table 3. Results of [27] without the use of AdaBoost

Models	α	μ
AE-CNN	99.96%	85.92%
SMOTE-AE-CNN	99.54%	45.40%
Naive Bayes [27]	97.71%	21.90%
Decision Tree [27]	99.92%	77.50%
Random Forest [27]	99.89%	60.40%
Gradient Boosted Tree [27]	99.90%	74.60%
Decision Stump [27]	99.90%	71.10%
Random Tree [27]	99.87%	49.70%
Deep Learning (MLP) [27]	99.92%	78.70%
Neural Network [27]	99.94%	81.20%
MLP [27]	99.93%	80.60%
Linear Regression [27]	99.91%	68.30%
Logistic Regression [27]	99.93%	78.60%
SVM [27]	99.94%	81.30%

Table 4. Results of [27] with the use of AdaBoost

Models	α	μ
AE-CNN	99.96%	85.92%
SMOTE-AE-CNN	99.54%	45.40%
(AdaBoost) Naive Bayes [27]	98.04%	23.50%
(AdaBoost) Decision Tree [27]	99.92%	77.50%
(AdaBoost) Random Forest [27]	99.89%	60.40%
(AdaBoost) Gradient Boosted Tree [27]	99.90%	74.70%
(AdaBoost) Decision Stump [27]	99.91%	71.10%
(AdaBoost) Random Tree [27]	99.87%	49.70%
(AdaBoost) Deep Learning (MLP) [27]	99.92%	76.50%
(AdaBoost) Neural Network [27]	99.93%	80.70%
(AdaBoost) MLP [27]	99.93%	80.60%
(AdaBoost) Linear Regression [27]	99.91%	68.60%
(AdaBoost) Logistic Regression [27]	99.93%	78.60%
(AdaBoost) SVM [27]	99.93%	79.60%

Table 5. Results of major voting in [27]

Models	α	μ
AE-CNN	99.96%	85.92%
SMOTE-AE-CNN	99.54%	45.40%
Decision Stump + Gradient Boosted Tree [27]	99.85%	34.30%
Decision Tree + Decision Stump [27]	99.85%	36.10%
Decision Tree + Gradient Boosted Tree [27]	99.92%	73.70%
Decision Tree + Naive Bayes [27]	99.93%	78.80%
Naive Bayes + Gradient Boosted Tree [27]	99.92%	74.20%
Neural Network + Naive Bayes [27]	99.94%	82.30%
Random Forest + Gradient Boosted Tree [27]	99.87%	46.80%

We observe that our multistage *Autoencoder - CNN* model performs better than the remaining schemes exposing the highest ζ . In terms of ϵ and δ metrics, our model is the second best after the CNN. In comparison with the models in [25], our model (without the SMOTE technique) performs better in terms of AUC's score, while in comparison with every model available in [27], our model performs better in terms of MCC and accuracy. The proposed model (SMOTE - AE - CNN) performs best in terms of ζ , which is the most important metric, considering that the purpose of the model is to detect fraudulent transactions, that comes however at a cost of a slightly increased number of normal transactions that are classified as frauds. In our case 372 normal transactions have been clas-

sified as fraudulent, out of 84.936 total transactions, i.e. 0.44% approximately. In other words, with our model a finance institution could detect much more fraudulent transactions, at the cost of a slight increased number of misclassified normal transactions, which in our case, remains relatively small.

6 Conclusions

In this study, we propose the combination of an Autoencoder with a Convolutional Neural Network (CNN) to predict financial fraud events. We target to adopt the proposed scheme in a streaming environment where numerous transactions are performed per day. We argue of the provision of a monitoring scheme at the edge of the network close to the location where transactions are realized. Our goal is to perform the initial step of fraud detection close to the source of data and any identified event will accompany the data towards the Cloud back end where a more complex statistical processing takes place. We meet the challenges coming from highly imbalanced datasets, adopting the SMOTE technique when training our model. We compare our scheme with a Support Vector Machine (SVM) and a simple CNN model that performs without reducing the dimensions into the training dataset. In addition, we compare our model with the results of 2 recently published papers, where the proposed models performs better in terms of MCC, accuracy and AUC score. Our experimental evaluation reveals that our model performs better, in terms of ζ , while in terms of ϵ & ζ is the second best after the CNN model. In the first place of our future research plans, it is to focus on a more complex model that will fully depict the distribution of the involved data even for the minimized number of features. A candidate solution is the adoption of variational autoencoders for performing the first step of our process, i.e., the dimensionality reduction.

References

1. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: A B7CEDGF HIB7PRQTSUDGQ ICWVYX HIB edCdSISIXvg5r CdQTw XvefCdS. In: Proceedings of the IEEE (1998)
2. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. **16**(February 2017), 321–357 (2002). <https://doi.org/10.1613/jair.953>. ISSN 10769757
3. Altman, E.I., Hotchkiss, E.: Corporate financial distress and bankruptcy (2005). ISBN 9780471691891. <https://doi.org/10.1002/9781118267806>
4. Cima Global: Fraud risk management: a guide to good practice. Chartered Institute of Management Accountants, pp. 1–80 (2008)
5. Prasad, N.R., Almanza-Garcia, S., Lu, T.T.: Anomaly detection. Comput. Mater. Continua **14**(1), 1–22 (2009). <https://doi.org/10.1145/1541880.1541882>. ISSN 15462218
6. Banerjee, A., Chitnis, U.B., Jadhav, S.L., Bhawalkar, J.S., Chaudhury, S.: Hypothesis testing, type I and type II errors. Ind. Psychiatry J. **18**(2), 127–131 (2009). <https://doi.org/10.4103/0972-6748.62274>

7. Bellotti, T., Crook, J.: Support vector machines for credit scoring and discovery of significant features. *Expert Syst. Appl.* **36**(2 PART 2), 3302–3308 (2009). <https://doi.org/10.1016/j.eswa.2008.01.005>. ISSN 09574174
8. BCBS: Basel Committee on Banking Supervision Basel III : International framework for liquidity risk measurement, standards and monitoring. Number December (2010). ISBN 9291318604
9. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.* **11**(110), 3371–3408 (2010)
10. Qi, M., Zhao, X.: Comparison of modeling methods for loss given default. *J. Bank. Financ.* **35**(11), 2842–2855 (2011). <https://doi.org/10.1016/j.jbankfin.2011.03.011>. ISSN 03784266
11. Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.* **39**(3), 3446–3453 (2012). <https://doi.org/10.1016/j.eswa.2011.09.033>. ISSN 09574174
12. Loterman, G., Brown, I., Martens, D., Mues, C., Baesens, B.: Benchmarking regression algorithms for loss given default modeling. *Int. J. Forecast.* **28**(1), 161–170 (2012). <https://doi.org/10.1016/j.ijforecast.2011.01.006>. ISSN 01692070
13. Harris, T.: Quantitative credit risk assessment using support vector machines: broad versus Narrow default definitions. *Expert Syst. Appl.* **40**(11), 4404–4413 (2013). <https://doi.org/10.1016/j.eswa.2013.01.044>. ISSN 09574174
14. Bastos, J.A.: Ensemble predictions of recovery rates. *J. Financ. Serv. Res.* **46**(2), 177–193 (2014). <https://doi.org/10.1007/s10693-013-0165-3>. ISSN 09208550
15. Pozzolo, A.D., Caelen, O., Le Borgne, Y.A., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Syst. Appl.* **41**(10), 4915–4928 (2014). <https://doi.org/10.1016/j.eswa.2014.02.026>. ISSN 09574174
16. Pozzolo, A.D.: Adaptive machine learning for credit card fraud detection - Dalpozzolo2015PhD.pdf, December 2015. <http://www.ulb.ac.be/di/map/adalpozz/pdf/Dalpozzolo2015PhD.pdf>
17. Pozzolo, A.D., Caelen, O., Johnson, R.A., Bontempi, G.: Calibrating probability with undersampling for unbalanced classification. In: Proceedings - 2015 IEEE Symposium Series on Computational Intelligence, SSCI 2015, no. November, pp. 159–166 (2015). <https://doi.org/10.1109/SSCI.2015.33>
18. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. *Comput. Secur.* **57**, 47–66 (2016). ISSN 0167-4048. <https://doi.org/10.1016/j.cose.2015.09.005>. <http://www.sciencedirect.com/science/article/pii/S0167404815001261>
19. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings, pp. 1–14 (2016)
20. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016). <http://www.deeplearningbook.org>
21. Barboza, F., Kimura, H., Altman, E.: Machine learning models and bankruptcy prediction. *Expert Syst. Appl.* **83**, 405–417 (2017). <https://doi.org/10.1016/j.eswa.2017.04.006>. ISSN 09574174
22. Carcillo, F., Dal Pozzolo, A., Le Borgne, Y.A., Caelen, O., Mazzer, Y., Bontempi, G.: SCARFF: a scalable framework for streaming credit card fraud detection with spark. *Inf. Fusion* **41**(September), 182–194, 2018. ISSN 15662535. <https://doi.org/10.1016/j.inffus.2017.09.005>

23. Dal Pozzolo, A., Boracchi, G., Caelen, O., Alippi, C., Bontempi, G.: Credit card fraud detection: a realistic modeling and a novel learning strategy. *IEEE Trans. Neural Netw. Learn. Syst.* **29**(8), 3784–3797 (2018). <https://doi.org/10.1109/TNNLS.2017.2736643>. ISSN 21622388
24. Fan, Q., Yang, J.: A denoising autoencoder approach for credit risk analysis (2018). <https://doi.org/10.1145/3194452.3194456>
25. Pumsirirat, A., Yan, L.: Credit card fraud detection using deep learning based on auto-encoder and restricted Boltzmann machine. *Int. J. Adv. Comput. Sci. Appl.* **9**(1), 18–25 (2018). <https://doi.org/10.14569/IJACSA.2018.090103>. ISSN 21565570
26. Zhu, B., Yang, W., Wang, H., Yuan, Y.: A hybrid deep learning model for consumer credit scoring. In: 2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018, no. May, pp. 205–208 (2018). <https://doi.org/10.1109/ICAIBD.2018.8396195>
27. Randhawa, K., Loo, C.K., Seera, M., Lim, C.P., Nandi, A.K.: Credit card fraud detection using AdaBoost and majority voting. *IEEE Access* **6**, 14277–14284 (2018). <https://doi.org/10.1109/ACCESS.2018.2806420>
28. Chen, J., Shen, Y., Ali, R.: Credit card fraud detection using sparse autoencoder and generative adversarial network. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference, IEMCON 2018, no. May, pp. 1054–1059 (2019). <https://doi.org/10.1109/IEMCON.2018.8614815>
29. Dupond, S.: A thorough review on the current advance of neural network structures. *Ann. Rev. Control.* **14**, 200–230 (2019)
30. Leo, M., Sharma, S., Maddulety, K.: Machine learning in banking risk management: a literature review. *Risks* **7**(1) (2019). ISSN 22279091. <https://doi.org/10.3390/risks7010029>
31. Wang, D., et al.: A semi-supervised graph attentive network for financial fraud detection. In: 2019 IEEE International Conference on Data Mining (ICDM) (2019). <https://doi.org/10.1109/ICDM.2019.00070>
32. Lebichot, B., Le Borgnee, Y.-A., He-Guelton, L., Oble, F., Bontempi, G.: Recent advances in big data and deep learning. In: Proceedings of the International Neural Networks Society, no. January, pp. 78–88 (2020). <https://doi.org/10.1007/978-3-030-16841-4>
33. Lebichot, B., Le Borgnee, Y.-A., He-Guelton, L., Oble, F., Bontempi, G.: Recent advances in big data and deep learning. In: Proceedings of the International Neural Networks Society, no. January, pp. 78–88 (2020). <https://doi.org/10.1007/978-3-030-16841-4>
34. Valueva, M.V., Nagornov, N.N., Lyakhov, P.A., Valuev, G.V., Chervyakov, N.I.: Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Math. Comput. Simul.* **177**, 232–243 (2020). ISSN 0378–4754. <https://doi.org/10.1016/j.matcom.2020.04.031>. <http://www.sciencedirect.com/science/article/pii/S0378475420301580>



Evolutionary Computation Approach for Spatial Workload Balancing

Ahmed Abubahia^{1(✉)}, Mohamed Bader-El-Den², and Ella Haig²

¹ School of Psychology and Computer Science, University of Central Lancashire,
Preston, UK

AAbubahia@uclan.ac.uk

² School of Computing, University of Portsmouth, Portsmouth, UK

Abstract. The growing demand for Geographic Information Systems (GIS) calls for high computation reliability to handle vast and complex spatial data processing tasks. A better parallel computing scheme should ensure balanced workload at different data processors to ensure optimal use of computing resources and minimise execution times, which poses more challenges with spatial data due to the nature of having spatial correlations and uneven distributions. In this paper, we propose a spatial clustering approach for workload balance, by using an evolutionary computation method that considers the nature of spatial data, to increase the computation performance for processing GIS polygon-based maps with massive number of vertices and complex shapes. To evaluate our proposed approach, We proposed two different experimental approaches for comparing our results: (i) Non-merging based experiment, and (ii) merging based experiment. The results demonstrated the advantage of the proposed spatial clustering approach in real GIS map based partitioning scenarios. The advantages and limitations of the proposed approach are discussed and further research directions are highlighted toward a development work by the research community.

Keywords: Computational optimisation · Geographic information system · Spatial data · Workload balancing · Evolutionary computation

1 Introduction

As geospatial information become immense in demand, more reliable approaches are needed for achieving better processing performance. Integrating parallel computing and spatial analysis tasks provides a promising solution to the complexity of GIS data processing [13, 15, 24, 31, 32, 35, 36, 38].

Geospatial data comprises two model types¹: raster data model and vector data model. The raster data represent geographical entities/features by a grid of intensity pixels. The best example of raster data is satellite images. In contrast,

¹ <http://www.esri.com/content/dam/esrisites/en-us/media/pdf/teach-with-gis/raster-faster.pdf>.

the vector data represent geographical entities/features by a set of vertices and paths. There are three different shapes to represent the vector data, they are:

- Point – useful for representing features at small scale (e.g. bus stops).
- Line/Polyline – useful for representing linear or curved features (e.g. roads or rivers).
- Polygon/Area – useful for representing features at large scale (e.g. map of country).

This paper focuses on the vector polygon type of GIS data. The best parallel computing scheme should ensure balanced workload at different data processors to ensure optimal use of computing resources (i.e. distribute the workload equally to all processors, rather than having some overloaded processors and some idle ones), which poses more challenges with GIS data [8, 16, 30, 36, 39].

In the context of GIS vector data, the most known implementation examples are GIS-Hadoop [2] and Spatial-Hadoop [10]. They have been introduced as potential solutions for parallel spatial data processing. Although these systems have shown a good performance in terms of spatial data storage and query processing, they still lack the partition strategy that meets the workload balancing requirement. A particular challenge in most spatial analysis tasks is that a map polygon should be processed as a united structure that consists of a set of vertices. Each vertex can be considered as a tuple in database terminology. The workload balancing, in case of GIS data, could be met by partitioning the GIS map into groups of polygons where these groups should be approximately equal in terms of the total number of vertices, which is an optimisation challenge. A common heuristic approach for optimisation is the use of evolutionary computation algorithms, of which the most popular is the Genetic Algorithm (GA) approach. In this paper, we argue that the spatial workload balancing challenge could be solved by applying evolutionary computation to partition large GIS maps into groups of polygons. These groups should be approximately equal in terms of the total number of vertices, and we propose an evolutionary computation based approach that consider both the nature of spatial data and the workload balancing requirement to extend the computation reliability for processing GIS complex data.

The rest of this paper is organised as follows: Sect. 2 reviews previous work on GIS vector map partitioning. Section 3 introduces the proposed evolutionary based approach for balanced workload based GIS vector map partitioning. Section 4 describes the experiments, including the data used and the experimental setup for the evaluation of the proposed partitioning approach. Section 5 discusses the experimental results, while Sect. 6 concludes the paper and outlines directions for future work.

2 Background and Related Work

Geospatial information systems (GIS) are computer-based systems that facilitate the input, storage, manipulation and output of geographic location-based

data [27]. GIS data models are classified into two categories: raster and vector data models. outlines the different properties of vector and raster data. In GIS context, satellite images are the most known example of the raster model. GIS vector data, which is the focus on this paper, has three components: spatial data, attribute data and index data. Spatial data describes the map itself and always takes the form of three basic geometrical entities, which are: points, lines/polylines and polygons. Points are used to define a single location of an object; they are used to represent real-world objects, such as bus stops, traffic lights and street lights. Lines/Polylines define linear objects; they can range from two-point lines to complex strings that have many vertices; lines are used to represent real-world objects, such as rivers and roads. Polygons define area-based objects; they can range from rectangles to multi-sided shapes with many vertices; polygons are used to represent real-world objects, such as lakes, shopping areas, buildings and city boundaries.

All these map entities are formed by many organised vertices; spatial data is actually a sequence of coordinates of these vertices based on a certain geographical coordinate system. The most used format of GIS spatial data is the ESRI (Environmental Systems Research Institute) shape file. The ESRI shape file [12] has become an industry standard in geospatial data due to its compatibility, to some extent, with recently released GIS software products. The attribute data describes the properties of map entities through links to the location data. Attributes can be, for example, names or matching addresses. The most known example of GIS attribute data format is the ESRI database file that is associated with the ESRI shape file and needs to have the same prefix as the shape file [12]. Last but not least, in the GIS context, the index data describes a file structure, such as total file length, for either spatial or attribute data. The ESRI index file [12] is the best known example of index files. Large regional partitioning is the process of dividing a large geographic area consisting of spatial objects, i.e. points, lines or polygons [22]. This paper focuses on the polygon type of map entities. Partitioning a large map into sub-sets of spatial entities is not an easy task due to the nature of having spatial correlations and uneven distribution. This problem has been investigated mostly in the redistricting field of GIS applications [4, 22, 28]. Some work has been done in the research of graph and GIS map clustering [5–7, 9, 11, 14, 20, 23, 26, 33, 34] and more attention given to the clustering of polygon-based type of GIS maps.

The previous approaches focus on attribute data rather than spatial data, and used evolutionary computation techniques for optimising the polygons' partitioning based on the attribute data, such as polygon area or polygon population [17–19, 21, 37].

According to the MapReduce model, the workload balancing can be only achieved by distributing equal chunks of data records (i.e. number of vertices) to the MapReduce processors [3, 8, 10, 25, 29].

Here the constraint is that the set of vertices that belong to the same polygon should not be separated in the mapping task (i.e. the first task of the MapReduce process). In contrast to the previous work, this paper focuses on spatial

properties of GIS vector data, and considers both the nature of spatial data and the workload balancing requirement to extend the computation reliability for processing GIS complex data. We propose an approach of workload balancing by using evolutionary computation in partitioning large GIS maps into groups of polygons which are approximately equal in terms of the total number of vertices (i.e. number of data records). We proposed two different experimental approaches for comparing our results: (i) Non-merging based experiment, and (ii) merging based experiment.

3 GIS Map Partitioning

This section outlines the main steps for implementing the proposed partitioning approach, including the map index computation, and applying the genetic algorithm to the problem of workload-balanced partitions in the context of spatial data.

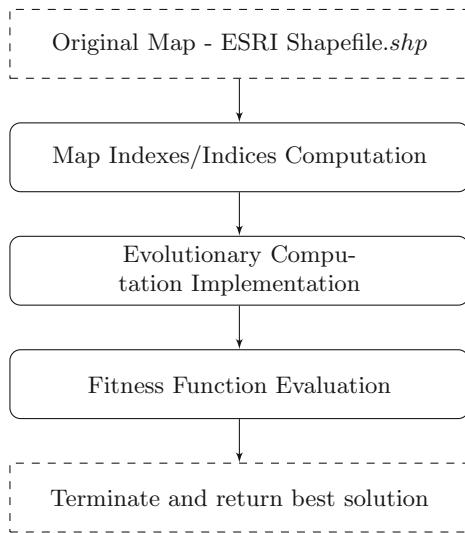


Fig. 1. The proposed evolutionary-based approach.

3.1 Map Indexes/Indices Computation

In the proposed approach (shown in Fig. 1), we argue that the use of polygons' representatives (indexes) will lead to faster processing rather than the use of the whole set of polygons' vertices. For identifying the map spatial features (polygons) indexes, we used the polygon's property of bounding box. Each polygon in

the GIS vector map has a defined bounding box, which identifies the boundaries of each polygon in the map; the coordinates for the bounding box are available in the shapefile [12]. The polygons' bounding box centres are calculated in both axes, as shown in Eq. (1) and Eq. (2), respectively.

$$x_c = \frac{x_{min} + x_{max}}{2} \quad (1)$$

$$y_c = \frac{y_{min} + y_{max}}{2} \quad (2)$$

where: x_c and y_c are the coordinates of polygon's centre in both x and y axes, respectively; x_{min} is the minimum vertex coordinate in x-axis; x_{max} is the maximum vertex coordinate in x-axis; y_{min} is the minimum vertex coordinate in y-axis; y_{max} is the maximum vertex coordinate in y-axis. x_{min} , x_{max} , y_{min} and y_{max} are each of 8-byte length [12].

Algorithm 1: Genetic Algorithm

Data: polygon based map; Seed Population(POP_Size); crossover rate; mutation rate

Result: near-optimal balanced map partitions

```

1 START;
2 Initiate  $POP (POP_{size})$  ;
3 Evaluate  $POP$  ;
4 while  $GEN \leq GEN_{size}$  do
5   for  $i \leftarrow 1$  to  $POP_{size} \times crossover_{ratio}$  do
6     | Parent1 = Tournament Selection ( $POP, T_{size}$  ;
7     | Parent2 = Tournament Selection ( $POP, T_{size}$  ;
8     | (Child1, Child2) = crossover(Parent1 , Parent2) ;
9   end
10  |  $POP_{new} \leftarrow Child_1$  ;
11  |  $POP_{new} \leftarrow Child_2$  ;
12  for  $i \leftarrow 1$  to  $POP_{size} \times mutation_{ratio}$  do
13    | Parent1 = Tournament Selection ( $POP, T_{size}$  ;
14    | Child1 = mutate(Parent1) ;
15  end
16  |  $POP_{new} \leftarrow Child_1$  ;
17  |  $POP \leftarrow POP_{new}$  ;
18  Evaluate  $POP$  ;
19  |  $GEN++$  ;
20 end
21 Print best evolution/solution ;
22 STOP;
```

3.2 Evolutionary Computation Implementation

The Genetic Algorithm (GA) optimization technique is based on random search and has many advantages, such as performing search in complex and large spaces, and providing near-optimal solutions. Unlike other optimization methods, GA is more suitable to this context of discrete variables based optimization problems. As shown in Algorithm 1, GA is a heuristic search algorithm that is based on evolutionary computation, which uses a random search to solve optimization problems. GA involves five main phases: initial population, fitness computation, selection, crossover and mutation. In Algorithm 1, POP refers to the population of individuals; POP_{size} represents the number of populations; GEN_{size} represents the number of generations; T_{size} is the number of tournaments. The initial population is randomly generated as a set of individuals, called a population, that represent solutions to the map partition problem. The parent selection is the process of selecting the fittest two pairs of individuals (i.e. candidate solutions), based on standard deviation value, to be used in producing the next generation. As shown in Fig. 2, the crossover operator is the process of mating the selected parents to produce the next generation by identifying a crossover point. One crossover point is selected, and the coordinate values after the crossover point are copied from the second parent to the first child, and from the first parent to the second child. The mutation operator is responsible for maintaining diversity within the population and preventing premature convergence. As shown in Fig. 3, the selected coordinate value is inverted to a new coordinate value. The parent selection, crossover and mutation processes are carried for both horizontal lines (X-axis) and vertical lines (Y-axis), as shown in Fig. 4.

41.8	42.5	43.3	Parent 1
42.1	42.7	43.2	Parent 2
41.8	42.7	43.2	Child 1
42.1	42.5	43.3	Child 2

Fig. 2. Crossover diagram example

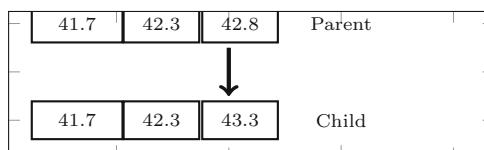


Fig. 3. Mutation diagram example

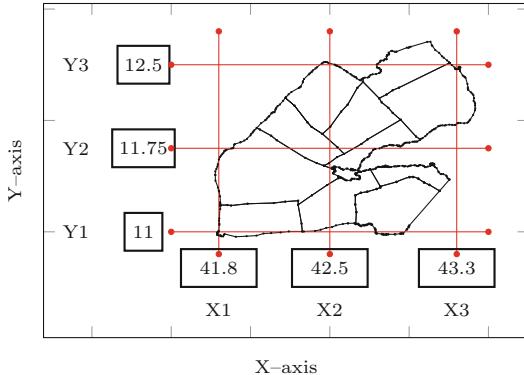


Fig. 4. Chromosome example of the horizontal and vertical solution lines (3×3 Lines or 4×4 cells)

In the proposed approach (Fig. 1), the resulted lines are combined to form the partitioning solutions, i.e. a set of three horizontal lines and three vertical lines would lead to sixteen (4×4) partitions. A collection of such solutions is called a population. The tournament selection method is used for selecting the parents, which has the advantage of diversifying the individuals set (i.e. candidate solutions). The process of parent selection, crossover and mutation continues iteratively for a specified number of generations until the fitness function is satisfied.

The fitness function for our problem is the standard deviation, as illustrated in Eq. (3). The best solution is defined by the minimum standard deviation value.

$$\sigma = \sqrt{\frac{\sum_{i=1}^N (x_i - \mu)^2}{N}} \quad (3)$$

where: σ is the standard deviation; x_i represents each value in the population; μ is the mean value of the population; and N is the number of values in the population.

The standard deviation is calculated at the level of the set of map partitions, where each set contain a number of polygons. The smaller the value of the standard deviation, the better the balance between the partitions according to the total number of vertices.

4 Data and Experiments

This section discusses the experimental setup for evaluating the performance and effectiveness of the proposed approach. Section 4.1 describes the data used in two sets of experiments; the initial set of experiments showed that some partitions had no vertices due to the uneven distribution of the data; consequently, an additional step was added to the partitioning approach to deal with these issues

and a second set of experiments was carried out. Section 4.2 describes the initial experiments, while Sect. 4.3 describes the second set of experiments.

4.1 Data and Materials

We implemented our proposed approaches (Fig. 1) on a PC machine with the following specification: Windows–10 home premium 64-bits operating system, CPU 2.5 GHz and RAM 8 GB. The programming tasks has been implemented with Java version 8 update 171 in Netbeans integrated development environment.

To allow comparisons for maps of different sizes in terms of number of polygons and number of vertices, four datasets (of two maps each) combining high and low numbers of polygons and vertices were used, respectively:

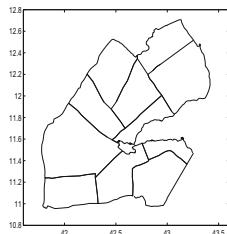
- **Dataset 1** includes maps with small number of polygons and small number of vertices.
- **Dataset 2** includes maps with small number of polygons and large number of vertices.
- **Dataset 3** includes maps with large number of polygons and small number of vertices.
- **Dataset 4** includes maps with large number of polygons and large number of vertices.

Within each dataset, the two maps are chosen to represent opposite ratios of number of polygons to number of vertices, i.e. one map has on average a smaller number of vertices per polygon compared with the other map in the same dataset. As shown in Table 1, eight GIS vector maps were used to implement our proposed approach, which are illustrated in Fig. 5, Fig. 6, Fig. 7 and Fig. 8.

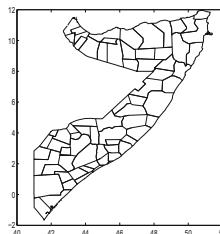
Table 1. The datasets with corresponding number of polygons, vertices and proportions of map size.

Dataset	Map	No. of polygons	No. of vertices	Average no. of vertices/polygon
1	Djibouti	11	676	61
	Somalia	88	3175	36
2	Guinea	56	21304	380
	Zimbabwe	81	32382	399
3	Liberia	305	10521	34
	Chad	347	19542	56
4	Burkina Faso	351	113996	324
	Ethiopia	575	261880	455

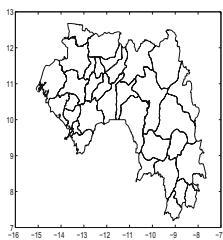
The used GIS maps are polygon-based maps that represent administrative boundaries of 8 countries in Africa, they are: Djibouti map of 11–polygons and



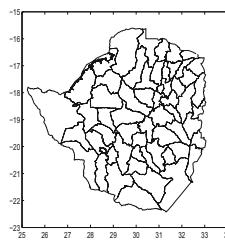
(a) Djibouti (11 polygons, 676 vertices)



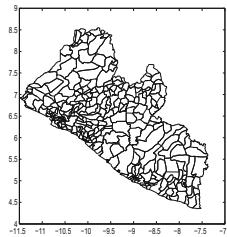
(b) Somalia (88 polygons, 3175 vertices)

Fig. 5. Data set 1.

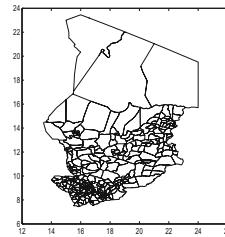
(a) Guinea (56 polygons, 21304 vertices)



(b) Zimbabwe (81 polygons, 32382 vertices)

Fig. 6. Data set 2.

(a) Liberia (305 polygons, 10521 vertices)



(b) Chad (347 polygons, 19542 vertices)

Fig. 7. Data set 3.

676–vertices (Fig. 5a), Somalia map of 88–polygons and 3175–vertices (Fig. 5b), Guinea map of 56–polygons and, 21304–vertices (Fig. 6a), Zimbabwe map of 81–polygons and 32382–vertices (Fig. 6b), Liberia map of 305–polygons and 10521–vertices (Fig. 7a), Chad map of 347–polygons and 19542–vertices (Fig. 7b), Burkina Faso map of 351–polygons and 113996–vertices (Fig. 8a) and Ethiopia map of 575–polygons and 261880–vertices (Fig. 8b). These vector maps are freely avail-

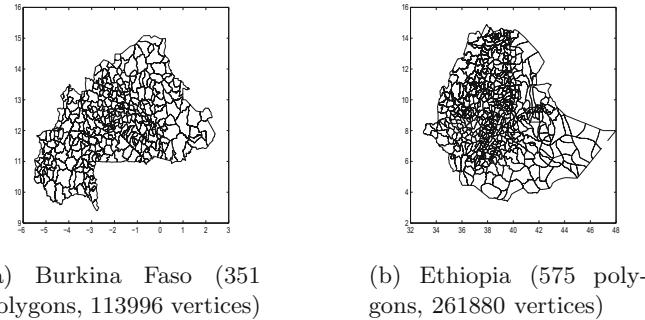


Fig. 8. Data set 4.

able, in ESRI shapefile format², from the Map Library website³. ESRI Shapefiles (.shp) are considered as a popular format for geographic information system applications [1]. They have several key features: supporting point, polyline and polygon geometry formats, fast shape editing, easy reading and writing, small storage space, and storing both spatial/attribute information [12].

We ran an initial experiment which is described in Sect. 4.2 in which we found that some partitions had no vertices, especially for maps containing concave shapes. To address this issue, we introduce a merging step at each iteration. Consequently, we refer to the first experiment as “non-merging” (described in Sect. 4.2) and the second experiment as “merging” (described in Sect. 4.3).

4.2 Non-merging Based Experiment

The experimental setup were as follows: both population size and generation number parameters were set to 10, 15 or 20, respectively. The crossover parameter was set to 0.8, the mutation parameter was set to 0.1, and the ratio parameter was set to 0.1. Moreover, each of these experiments was ran for 51 times – an odd number was chosen to have a median value as a data point rather than an average of the middle two data points. The standard deviation is used as the fitness function. In our experiments, the number of grid cells (i.e. the number of partitions) were selected according to the number of polygons. For more clarification, the number of cells was set to 4×4 for maps with small number of polygons (i.e. maps of Djibouti, Somalia, Guinea and Zimbabwe), while the number of cells was set to 6×6 for maps with a large number of polygons (i.e. maps of Liberia, Chad, Burkina Faso and Ethiopia). The number of partitions can be user-defined to match the available number of processors in systems like MapReduce.

² <http://www.esri.com>.

³ <http://www.maplibrary.org/library/stacks/Africa/index.htm>.

4.3 Merging Based Experiment

This approach follows the same implementation steps as in the non-merging experiment that were given above. The only difference is that before computing the fitness function, a merging procedure is applied to the partitions based on a threshold value. In this paper, the average number of vertices per polygon is used as threshold value. This will be advantageous in avoiding the partitions that contain no vertices, i.e. the number of vertices is equal to zero.

The threshold value (i.e. the number of vertices per cell) were selected according to the average number of vertices per polygon, and then set as follows: Djibouti Map (61 vertices), Somalia Map (36 vertices), Guinea Map (380 vertices), Zimbabwe Map (399 vertices), Liberia Map (34 vertices), Chad Map (56 vertices), Burkina Faso Map (324 vertices), and Ethiopia Map (455 vertices).

5 Results and Discussion

In this section the experimental results of the two sets of results are presented and discussed. Figure 9 shows a solution example for the Djibouti map for both the non-merging and merging-based partitioning results for the experimental settings of: 51 run times, population size parameter of 20, generation number parameter of 20, crossover parameter of 0.8, mutation parameter of 0.1, and the ratio parameter of 0.1.

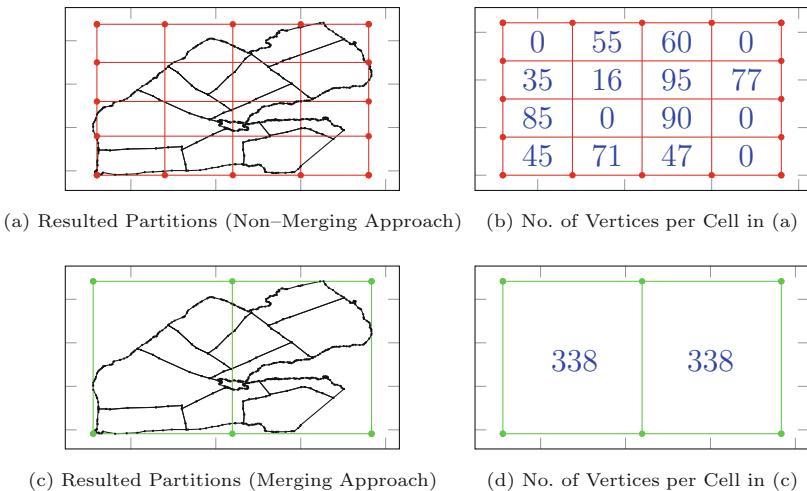


Fig. 9. Djibouti map, examples of the best solution (chromosome)

In the non-merging experiment, as shown in the Fig. 9(a), there are 16-cells that represent the resulted partitions. Two partitions, i.e. upper-left and upper-right corner cells, contain no vertices. Figure 9(b) illustrates clearly how

the number of vertices are distributed over the resulted partition based number of grid cells. In the merging experiment, as shown in the Fig. 9(c), there are 2 cells that represent the resulted partitions which should contain equal or nearly-equal number of vertices. In the shown figure each cell contains 338 vertices. Figure 9(d) illustrate clearly how the number of vertices are distributed over the resulted partition based number of grid cells. To compare the results of the two sets of experiments, as well as the influence of the different values for the population size and the number of generations, we present the results in the form of box plots illustrating the range of values for the fitness function, i.e. the standard deviation. A box plot is a graphical shape for displaying the statistical range of values. Beginning from the top, the upper whisker represents the highest value in the range. Seventy-five percent (75%) of the resulted values fall below the upper quartile. The median marks the middle-point of the resulted standard deviation values and is shown by the line that divides the box into two parts. Half of the resulted values are greater than or equal to this value and half of the results have values lower than the median. Twenty-five percent (25%) of the resulted standard deviation values fall below the lower quartile. Finally, the lower whisker represents the smallest value in the range.

Figure 10 and 11 display the results for Dataset 1 against the population sizes of 10 and 20, respectively. Similarly, Fig. 12 and 13 display the results for Dataset 2. Figure 14 and 15 illustrate the results for Dataset 3. Figure 16 and 17 show the results for Dataset 4.

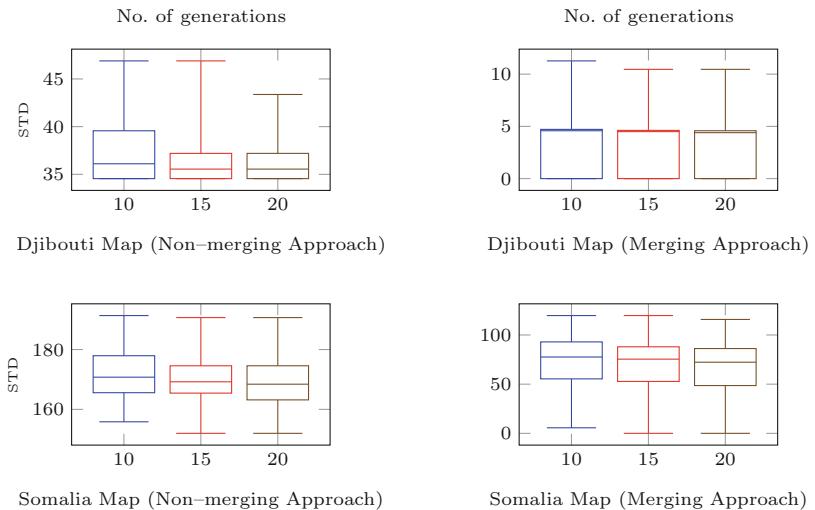


Fig. 10. Dataset 1, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 10

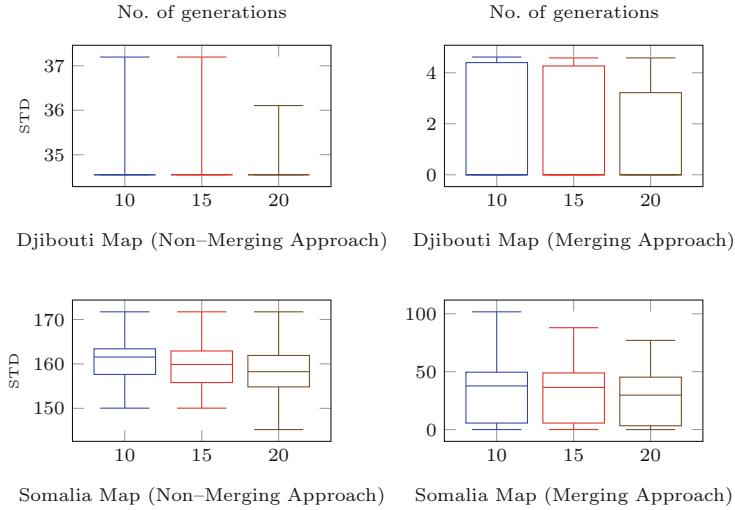


Fig. 11. Dataset 1, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 20

The experimental results show that an increase in the population size leads to lower values for the fitness criteria (standard deviation), which indicates better solutions, i.e. a more even distribution of the vertices among the partitions. Experiments with the Djibouti map, for example, show that when using the population size of 10, the standard deviation (STD) values range between 34.5 and 46.8 for the non-merging approach (top left in Fig. 10), and between 0 and 11.2 for the merging approach (top right in Fig. 10). For the population size of 20, the standard deviation values range between 34.5 and 37.1 for the non-merging approach (top left in Fig. 11), and between 0 and 4.5 for the merging approach (top left in Fig. 11).

For all population sizes (10 or 20 populations), the results show that the higher the generations number (10, 15 or 20 generations) for the reproduction process, the better the solutions produced, i.e. lower values for the standard deviation. This applies to all datasets regardless of the number of polygons or the number of vertices. Non-merging based experiments with the Liberia map (dataset 3), for example, show that in non-merging based experiments with population size of 20, when reproducing for 10 generations for the non-merging approach, the standard deviation values range between 203 and 318.6; for 15 generations, the standard deviation values are reduced to the range between 203 and 307.6; for 20 generations, the range is further reduced between 203 and 292.3; (top left in Fig. 15).

While when experimenting with the same map, show that in merging based experiments with population size of 20, when reproducing for 10 generations for the non-merging approach, the standard deviation values range between 116.6 and 284.4; for 15 generations, the standard deviation values are reduced

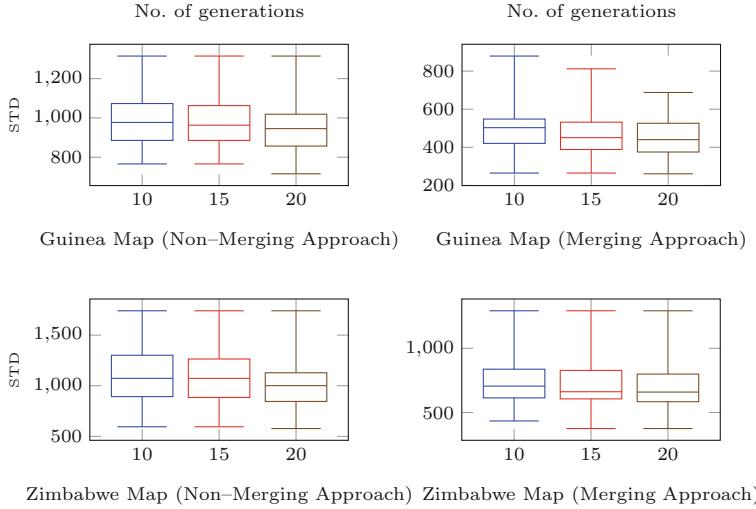


Fig. 12. Dataset 2, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 10

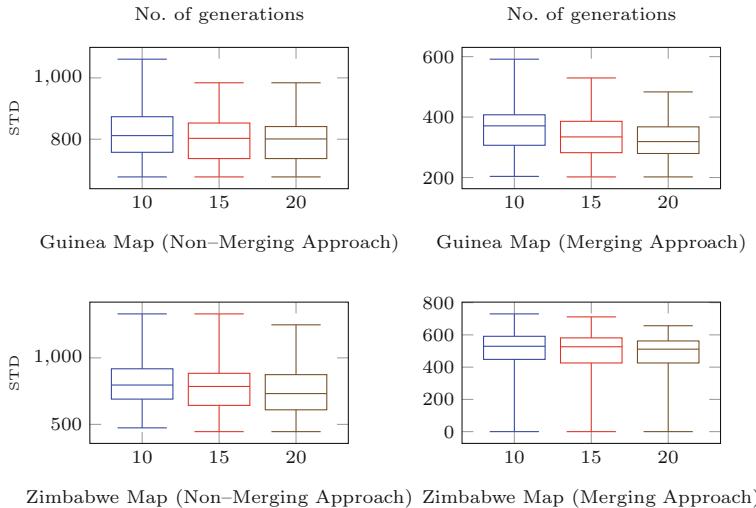


Fig. 13. Dataset 2, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 20

to the range between 116.6 and 257.7; for 20 generations, the range is further reduced between 116.6 and 245.8; (top right in Fig. 15). When comparing both experiments with population size of 20 and generation size of 20, it can be seen that the merging based experiment showed better standard deviation values than the non-merging based experiment. for example, for the map of Ethiopia

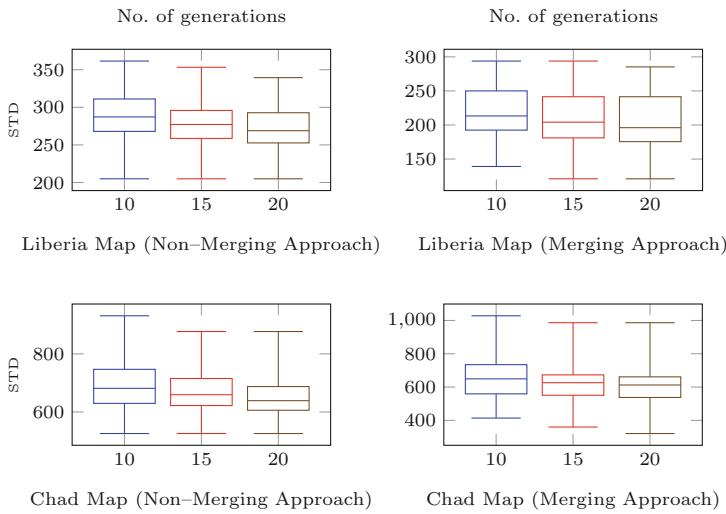


Fig. 14. Dataset 3, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 10

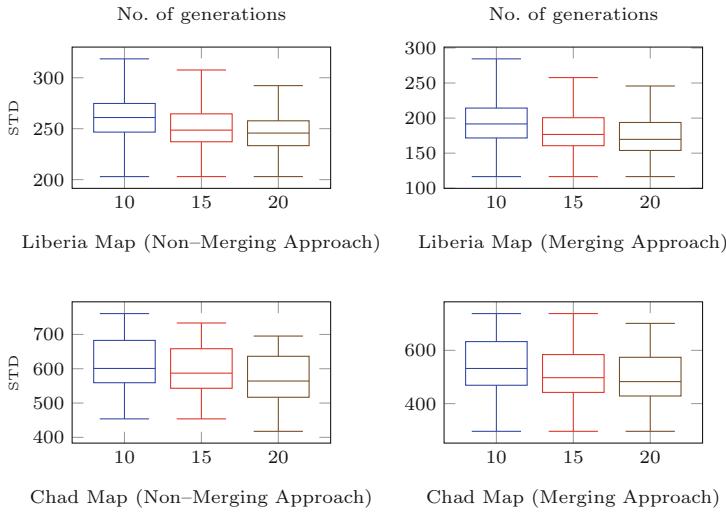


Fig. 15. Dataset 3, Comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 20

(Fig. 17, dataset 4), for the non-merging approach the standard deviation values range between 4895.7 and 9414.4 , while for the merging approach the range of the standard deviation values is between 3766.6 and 7757.7. All mentioned trends – i.e. the results improve with increasing population size, the results improve with increasing numbers of generations and the results improve in the

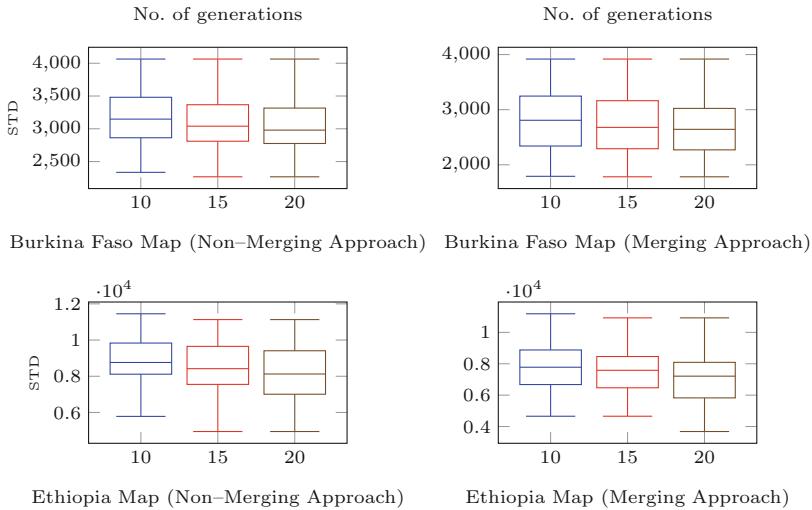


Fig. 16. Dataset 4, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 10

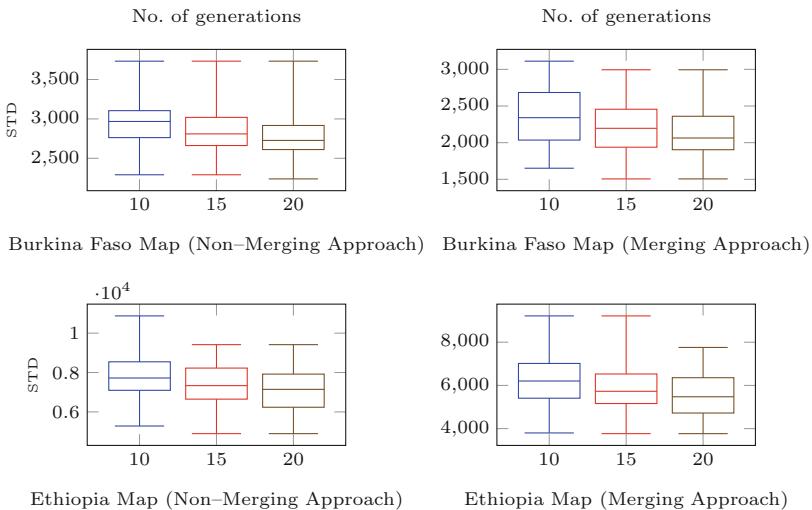


Fig. 17. Dataset 4, comparison between non-merging approach (left column) and merging approach (right column), no. of runs = 51, pop size = 20

merging approach compared with the non-merging approach – can be observed for all datasets: Dataset 1 (Fig. 10 and 11), Dataset 2 (Fig. 12 and 13), Dataset 3 (Fig. 14 and 15) and Dataset 4 (Fig. 16 and 17).

6 Conclusion and Further Research

In this paper, we discussed and highlighted the importance of spatial workload balancing for GIS vector map partitioning. To address this problem, we proposed an evolutionary-based approach for GIS map partitioning using the Genetic Algorithm (GA). Our proposed approach considers the nature of spatial data to increase the computation performance for processing GIS polygon-based maps with massive number of vertices and complex shapes. Four datasets were used, where each dataset had varying degrees of size in terms of number of polygons and number of vertices. Each dataset contained two maps, which had opposite ratios of number of vertices per polygon. A set of experiments on the four datasets were implemented to assess the influence of the evolutionary genetic algorithm parameters including the population size and the number of generations. The results showed the advantage of the proposed spatial workload balancing approach in real GIS map based partitioning scenarios. The use of evolutionary computation shows a promising potential in partitioning GIS maps into spatially balanced set of adjacent polygons based on the number of vertices.

The proposed approach in this paper is a first step toward a developed spatial workload balancing approach for GIS vector maps. Further research and experiments will be carried out on addressing the problem of the randomness in the map polygon shapes (concave and convex shapes) to further understand the behavior of the spatial workload balancing approach in extreme cases. Also, the possibility of introducing different weights for the different topological aspects will be investigated.

References

1. GIS (Geographic Information System) Overview. <https://www.esri.com/en-us/what-is-gis/overview>
2. Aji, A., et al.: Hadoop-GIS: a high performance spatial data warehousing system over MapReduce. In: The 39th International Conference on Very Large Data Bases, vol. 6, pp. 1009–1020 (2013)
3. Araujo Neto, A.C., Coelho da Silva, T.L., de Farias, V.A.E., Macêdo, J.A.F., de Castro Machado, J.: G2P: a partitioning approach for processing DBSCAN with MapReduce. In: Web and Wireless Geographical Information Systems, pp. 191–d-202. Springer, Cham (2015)
4. Bação, F., Lobo, V., Painho, M.: Applying genetic algorithms to zone design. Soft. Comput. **9**(5), 341–348 (2005)
5. Barua, H.B., Das, D.K., Sarmah, S.: A density based clustering technique for large spatial data using polygon approach. J. Comput. Eng. **3**(6), 1–9 (2012)
6. Boobalan, M.P., Lopez, D., Gao, X.: Graph clustering using k-neighbourhood attribute structural similarity. Appl. Soft Comput. **47**, 216–223 (2016)
7. Cao, Z., Wang, S., Forestier, G., Puissant, A., Eick, C.F.: Analyzing the composition of cities using spatial clustering. In: Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing, pp. 141–148 (2013)
8. Dean, J., Ghemawat, S.: MapReduce: simplified data processing on large clusters. In: the 6th Conference on Symposium on Operating Systems Design and Implementation, pp. 1–13. Google, Inc. (2004)

9. Eldawy, A., Alarabi, L., Mokbel, M.F.: Spatial partitioning techniques in Spatial-Hadoop. *Proc. VLDB Endow.* **8**(12), 1602–1605 (2015)
10. Eldawy, A., Mokbel, M.F.: SpatialHadoop: a MapReduce framework for spatial data. In: The 31st International Conference on Data Engineering, pp. 1352–1363 (2015)
11. Ericsson, A., WCDMA, R.: Clustering and polygon merging algorithms for finger-printing positioning in LTE. In: 5th International Conference on Signal Processing and Communication Systems (ICSPCS), pp. 1–10 (2011)
12. ESRI: ESRI shapefile technical description. Technical report, Environmental Systems Research Institute Inc, 380 New York Street, Redlands, CA 92373–8100, USA (1998)
13. Fu, Y.X., Zhao, W.Z., Ma, H.F.: Research on parallel DBSCAN algorithm design based on MapReduce. In: Advanced Measurement and Test, Advanced Materials Research, vol. 301, pp. 1133–1138. Trans Tech Publications (2011)
14. Gu, X., Angelov, P.P., Príncipe, J.C.: A method for autonomous data partitioning. *Inf. Sci.* **460–461**, 65–82 (2018)
15. Guest, O., Kanayet, F.J., Love, B.C.: Gerrymandering and computational redistricting. *J. Comput. Soc. Sci.* **2**(2), 119–131 (2019). <https://doi.org/10.1007/s42001-019-00053-9>
16. Gufler, B., Augsten, N., Reiser, A., Kemper, A.: The partition cost model for load balancing in MapReduce, chap. 5, pp. 371–387. Springer, New York (2012)
17. Jasim, M., Asadi, T.A.: New graph mining algorithm for vector GIS systems. In: 8th International Conference on Computing Technology and Information Management (NCM and ICNIT), vol. 1, pp. 335–338 (2012)
18. Ji, G., Zhang, L.: A spatial polygon objects clustering algorithm based on topological relations for GML data. In: 2009 International Conference on Information Engineering and Computer Science, pp. 1–4 (2009)
19. Joshi, D., Samal, A., Soh, L.K.: A dissimilarity function for clustering geospatial polygons. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 384–387. ACM, New York (2009)
20. Joshi, D., Samal, A.K., Soh, L.K.: Density-based clustering of polygons. In: 2009 IEEE Symposium on Computational Intelligence and Data Mining, pp. 171–178 (2009)
21. Joshi, D., Soh, L.K., Samal, A.: Redistricting using heuristic-based polygonal clustering. In: 2009 Ninth IEEE International Conference on Data Mining, pp. 830–835 (2009)
22. Joshi, D., Soh, L.K., Samal, A.: Redistricting using constrained polygonal clustering. *IEEE Trans. Knowl. Data Eng.* **24**(11), 2065–2079 (2012)
23. Kisore, N.R., Koteswaraiah, C.B.: Improving ATM coverage area using density based clustering algorithm and voronoi diagrams. *Inf. Sci.* **376**, 1–20 (2017)
24. Levin, H.A., Friedler, S.A.: Automated congressional redistricting. *J. Exp. Algorithmics* **24**, 1–24 (2019)
25. Li, X., Li, W., Anselin, L., Rey, S., Koschinsky, J.: A MapReduce algorithm to create contiguity weights for spatial analysis of big data. In: Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, pp. 50–53. ACM, New York (2014)
26. Liu, R., Wang, H., Yu, X.: Shared-nearest-neighbor-based clustering by fast search and find of density peaks. *Inf. Sci.* **450**, 200–226 (2018)
27. Longley, P.A., Goodchild, M., Maguire, D.J., Rhind, D.W.: *Geographic Information Systems and Science*, 3rd edn. Wiley, Hoboken (2011)

28. Photis, Y.N.: Redefinition of the Greek electoral districts through the application of a region-building algorithm. MPRA Paper 42398, University Library of Munich, Germany (2012)
29. Puri, S., Agarwal, D., He, X., Prasad, S.K.: MapReduce algorithms for GIS polygonal overlay processing. In: 2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and PHD Forum, pp. 1009–1016 (2013)
30. Qiu, Q., Yao, X., Chen, C., Liu, Y., Fang, J.: A spatial data partitioning and merging method for parallel vector spatial analysis. In: 2015 23rd International Conference on Geoinformatics, pp. 1–5 (2015)
31. Schutzman, Z.: Trade-offs in fair redistricting. In: AAAI/ACM Conference on AI, Ethics, and Society, pp. 159–165 (2020)
32. Shuliang, W., Gangyi, D., Ming, Z.: Big spatial data mining. In: IEEE International Conference on Big Data, pp. 13–21 (2013)
33. Wang, S., Chen, C.S., Rinsurongkawong, V., Akdag, F., Eick, C.F.: A polygon-based methodology for mining related spatial datasets. In: Proceedings of the 1st ACM SIGSPATIAL International Workshop on Data Mining for Geoinformatics, pp. 1–8 (2010)
34. Wang, S., Eick, C.F.: A polygon-based clustering and analysis framework for mining spatial datasets. *GeoInformatica* **18**(3), 569–594 (2013). <https://doi.org/10.1007/s10707-013-0190-2>
35. Wang, W., Du, S., Guo, Z., Luo, L.: Polygonal clustering analysis using multilevel graph-partition. *Trans. GIS* **19**(5), 716–736 (2015)
36. Wei, H., et al.: A kd tree-based algorithm to parallelize kriging interpolation of big spatial data. *GISci. Remote Sens.* **52**(1), 40–57 (2015)
37. Zhang, J., Samal, A., Soh, L.: Polygon-based spatial clustering. In: The 8th International Conference on GeoComputation, pp. 1–5 (2005)
38. Zhang, X., Huang, B., Tay, R.: Estimating spatial logistic model: a deterministic approach or a heuristic approach? *Inf. Sci.* **330**, 358–369 (2016). SI Visual Info Communication
39. Zhao, L., Chen, L., Ranjan, R., Choo, K.-K.R., He, J.: Geographical information system parallelization for spatial big data processing: a review. *Clust. Comput.* **19**(1), 139–152 (2015). <https://doi.org/10.1007/s10586-015-0512-2>



Depth Self-optimized Learning Toward Data Science

Ziqi Zhang^(✉)

School of Life Science, Tsinghua University, Beijing, China
zhangzq20@mails.tsinghua.edu.cn

Abstract. We propose a two-stage model called Depth Self-Optimized Learning (DSOL), which aims to realize ANN depth self-configuration, self-optimization as well as ANN training without manual intervention. In the first stage of DSOL, it will configure ANN of specific depths according to a specific dataset. In the second stage, DSOL will continuously optimize ANN based on Reinforcement Learning (RL). Finally, the optimal depth is returned to the first stage of DSOL for training, so that DSOL can configure the appropriate ANN depth and perform more reasonable optimization when processing similar datasets again. In the experiment, we ran DSOL on the Iris and Boston housing datasets, and the results showed that DSOL performed well. We have uploaded the experiment records and code to our Github (<https://github.com/workharduwillwin/Depth-Self-Optimized-Learning-Toward-Data-Science>).

Keywords: Depth self-optimized learning · Data science

1 Introduction

Date back to the 1989s, the universal approximation theorem states that a fixed depth neural network with arbitrary width and specific activation function such as sigmoid could approximate any continuous functions on a compact set to arbitrary accuracy [1, 2]. Approximate arbitrary functions might be the key point for a universal model, but when using this shallow ANN dealing with complex data, the number of its parameters will reach exponential level, so this method is difficult to implement. Subsequently, people found that increasing the depth of neural network can make neural network easily approximate functions on some datasets that shallow neural network can't [3–6]. Similar viewpoint could be found in Goodfellow's book—Deep learning [7] which states that in some cases, deeper networks can generalize better, and this is not just because of the larger number of parameters. Those views seems to be confirmed in real life, from AlexNet to RestNet, it is obvious that the performance of ANN is positively related to its depth, so it seems that only if we continuously increase the depth then we can gain a universal model.

Unfortunately, according to the No Free Lunch theorem (NFL) [8], no model can be fully qualified for various tasks, which seems to contradict our wishes.

Even if we design a general model for a variety of different datasets within the allowable error range, we do not need to use more complex methods for those tasks can be completed by simple methods. Because it may waste too much computing resources. However, if we try to understand the NFL from another perspective, it seems that the theorem itself can be regarded as an excellent way to obviously improve the generalized ability of ANN, that is to say, using the best ANN according to specific dataset. *In oreder to realize a universal mode based on the idea mentioned above, we mainly focus on solving those problems summarized as below:*

1. How to design ANNs according to specific datasets.
2. How to implement this design process through a model that can work independently.

Before this work, the above-mentioned second problem has been studied for a long time. In the past, people has realized Neural Architecture Search (NAS) based on RL, evolutionary algorithms, and so on [9]. Although the original intention of this type of research is to help people better search the hyperparameters of ANN, from another perspective, it is also an excellent way to realize ANN self-tuning. The first problem mentioned above may be more challenging, but we have gained some inspiration from human behavior, namely, we are trying to make ANNs understand what they will do. Based on these ideas and researches, we propose a two-stage model called Depth Self-Optimized Learning (DSOL). In the experiment, we used the Boston housing and Iris dataset to test the first stage of the DSOL. We set the True labels to 3, 10, 25, 50, 60, and train the first-stage for 100 iterations in turn. Experimental results show that the first stage of the DSOL can converge well on those two datasets, which indicates that the first-stage of DSOL has a excellent approximation and generalization capability to some extent. Furthermore, when we normalize the training data, we find that the predictions of the model becomes more accurate. Then, we test the second stage of DSOL. Significantly, due to the limited performance of the device, we can only set the maximum number of ANN layers to 15, that is to say, if the prediction is greater than 15, then it will be regarded as 15. After training, the second stage of DSOL and ANN converged. In addition, by visualizing the number of ANN layers, it can be observed that well-trained DSOL tends to choose more deep ANNs to approximate on the dataset. *Our contributions summarized as below:*

1. We proposed a two-stage model called self-optimization learning (DSOL) for data science, which initially realizes ANN depth self-configuration and self-optimization. The advantage of this model is that it brings the datasets into the ANN parameter configuration process to know what they are doing.

Although we have achieved staged victories, we still have a long way to go to achieve our ultimate goal. Our ultimate goal is to design a system that can automatically, accurately, and quickly generate, train, optimize ANNs for various specific datasets so as to realize a real universal model, namely, we only

have to input dataset into it without doing any other things. In addition, this system can utilize various potential functions and modules, and can continuously upgrade the ANN in consideration of the performance of ANN and the feature of the dataset rather than only the depth of ANN. *Now our model still has many limitations:*

1. The second stage(RL-stage) of DSOL may fall into the cycle of local minimum in some cases. As the matter of fact, we have considered this problem, so we add a little randomness to the decision-making process of the second stage, and solve this problem to some extent, but it is not very efficient.
2. If some ANNs with the depth have an obvious different performance on the same dataset, the judgment of the RL model will be affected. However, the parameters of ANN should be initialized with an appropriate method to ensure it could attend the optimal level, namely, if we set all parameters to 0, the performance of ANN will be affected, which is contrary to our goal. So we must find a balance between the stability of RL and the performance of ANN.
3. The current architecture of the DSOL is not very complicated, so although it can be used on small-dimensional dataset, its performance on large-scale dataset is still unknown. At the same time, we know that the performance of ANN is not only related to its depth, so we need to further improve the model so that it can configure and optimize more kinds of ANN parameters.

In the introduction, we have briefly discussed the research background, solutions, experiments, contributions, and limitations of this research. The rest of this paper is as follows: In Sect. 2 of this paper, we will briefly introduce some researches related to this work. In Sect. 3, we will introduce in detail the principles and architecture of DSOL. In Sect. 4, we will analyze the experimental results. In Sect. 5, we made a summary and stated our next research plan, and the last part is acknowledgment. Significantly, most of the experimental data and figures are in the attachment.

2 Related Work

In the past five years, Reinforcement Learning (RL) has solved many problems that is difficult for traditional machine learning (ML). For example, RL has reached a superhuman level in Atari game [10] and poker game [11]. In addition, RL has some practical applications, such as self-driving cars [12], and so on. In this paper, we mainly introduce its application in Neural Architecture Search (NAS).

Bbarret zoph et al. [13] used RCNN which is optimized by Reinforcement Learning (RL) based on policy gradient method to search for the best hyperparameters of ANN. They did it, but they used a lot of GPUs and ran for about a month, which was beyond the experimental conditions of ordinary people. Barret Zoph et al. [14] proposed another method based on RL. They did not search for the complete ANN architecture, but first constructed a cell architecture, and

then obtained an optimal ANN architecture constructed by these cell architectures. They used 500 GPUs and ran for about 4 days. These methods require a lot of computing resources, but there are some studies that have gradually reduced the requirement of computing source, such as some NAS researches based on Hierarchical Representation [15], Weight sharing [16], Performance prediction, and so on.

It is no doubt that NAS-related researches will make it more convenient for people to configure and optimize the hyperparameters of ANN. Our goal is not exactly the same as NAS, because we focus on developing a self-optimizing model rather than only help people configure hyperparameters of ANN. But we have some similarities, such as looking for a better ANN architecture.

3 Methods

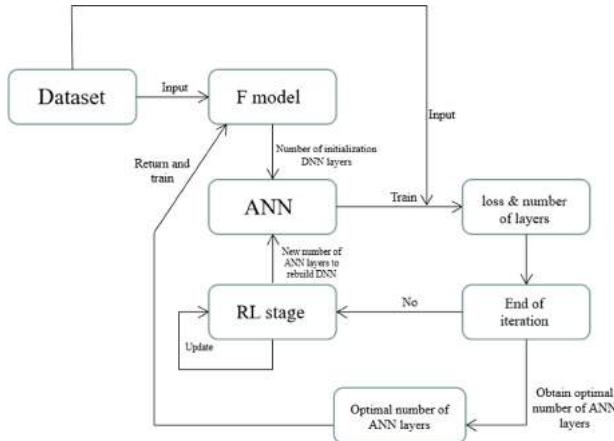


Fig. 1. The self-optimization learning model.

As shown in Fig. 1, Algorithm 1, DSOL includes two stages. In the first stage, the training dataset is input into the F-model to calculate the number of ANN layers, and then the RL model will optimize the number of layers according to the performance of the ANN. Significantly, ANN performance is related to its hyperparameters, thus a deep ANN doesn't mean must be better than a shallow one. However, using some methods such as Batchnormalization or Dropout, and use Relu or LeakyRelu to replace sigmoid or no activation function will efficiently solve the problem of gradient disappearance so as to improve the performance of ANN. In this paper, we use fixed 500 width Dense layer with Relu activation function, and every Dense layer followed by a Dropout layer to build ANN. Significantly, we initialize the bias of each Dense layer as 0, use Glorot Uniform methods to initialize the weights of ANN layers, and in the subsections, we will introduce the architecture and function of the DSOL in detail.

Algorithm 1. Depth Self-Optimized Learning

Initialize the state-policy function π weightens with Glorot Uniform method.
 Initialize the policy-value function Q weightens with Glorot Uniform method.
 Initialize replay buffer h_t to capacity N .
 Initialize the $F - model$ weights with Glorot Uniform method.
 Initialize the minimum loss value $minloss$ to a large interger.
for episode = 1, M **do**
 Initialize the number of ANN layers by $F model$, build a neural network ANN_t
 Train ANN in fixed iterations I , return loss value $loss_t$.
 for $t = 1, T$ **do**
 With probability ε select a random action a_t (Add or reduce a Dense layer).
 Otherwise select $a_t \leftarrow argmax(\pi(loss_t, layer_t; \theta))$.
 Update $layer_t$ to $layer_{t+1}$ according to a_t .
 Rebuild the neural network to ANN_{t+1} according to $layer_{t+1}$.
 Train ANN_{t+1} in fixed iterations I , return loss value $loss_{t+1}$.
 if $loss_{t+1} < minloss$ **then**
 $minloss \leftarrow loss_{t+1}$
 end if
 $r_t = \begin{cases} \frac{loss_t - loss_{t+1}}{loss_{t+1}} \times 10 + \frac{minloss - loss_{t+1}}{minloss} \times 10 & \text{if } loss_{t+1} > loss_t \\ \frac{loss_t - loss_{t+1}}{loss_t} \times 10 & \text{if } loss_{t+1} < loss_t \end{cases}$
 Store transition $(loss_t, layer_t, loss_{t+1}, layer_{t+1}, r_t)$ in h_t .
 Sample random minibatch $(loss_t, layer_t, loss_{t+1}, layer_{t+1}, r_t)$ from h_t .
 Update the parameters of the actor based on equation 4.
 Update the parameters of critics according to equation 5.
 end for
 Return the optimal number of the ANN layers.
 Update the parameters of the $F - model$ based on gradient descent.
end for

3.1 F-Model

The first stage of DSOL is F-model (Fig. 2, Table 1, Table 2). F-model consists of two convolutional layers, two pooling layers, and three or four fully-connected(FC) layers in turn. Significantly, according to the code we provided, there is a judgment in front of the FC layer that if necessary, add a Dense layer of appropriate width between the first FC and the last CNN layers so that the model can be successfully trained on various datasets (Table 1, Table 2). In addition, although the architecture of the F-model has been shown in this paper, we can use more powerful modules to replace it, such as combining RL with more powerful ANN to replace its current architecture. However, In order to successfully implement the ideas proposed in this paper, we will still use the initial design.

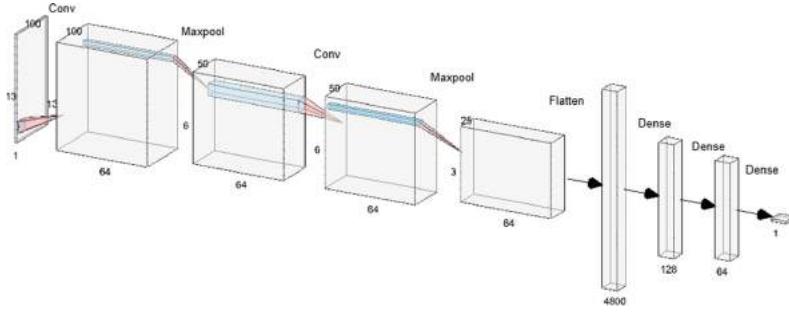


Fig. 2. Architecture of the F-model. *Using 100 samples from Boston dataset as training dataset, the architecture of F-model is shown in this figure.*

3.2 Reinforcement Learning Stage

Now, we introduce the architecture of the Reinforcement Learning (RL) stage. The principle of the RL stage is similar to the agent-environment interaction model. At the beginning of the t round of training, the environment will construct ANN_t according to the number of ANN_t layers $layer_t$ which is configured by the F-model. After ANN_t training, we obtain $loss_t$. In the experiment, both $loss_t$ and $layer_t$ are regarded as the state of the agent. Input the state $loss_t$ and $layer_t$ into the policy function ω , and then we can get the next action a_t (Eq. 1). Optimize ANN_t according to a_t so that obtain $layer_{t+1}$ (Eq. 2). Construct ANN_{t+1} according to $layer_{t+1}$, after ANN_{t+1} training, we will get a new loss value $loss_{t+1}$. If $loss_{t+1}$ is greater than $loss_t$, the action will be punished, on the contrary, the action will be rewarded. Regardless of punishment or reward, all of these can be represented by r_t . Then, we will obtain the sequence data h_t (Eq. 3), which will be used to train the policy function ω and the value function Q .

$$a_t = \omega(h_t; \beta) \quad (1)$$

$$layer_{t+1} = layer_t + a_t \quad (2)$$

$$h_t = \{loss_1, layer_1, r_1, loss_2, layer_2, r_2, \dots, loss_t, a_t, r_t\} \quad (3)$$

τ is a constant decay rate range from 0 to 1 (Here we use 0.8), and r_t is a reward based on the performance of ANN_t . Therefore, the total reward in the RL training stage can be represented by the reward function: $R_t = \sum_t^\infty \tau^{t'-t} r_{t'}$. In order to maximize R_t , we use the policy gradient (Eq. 4) to optimize the policy function ω , and use the TD error (Eq. 5) to optimize the value function Q . Finally, we will obtain the optimal policy function ω^* , and the value function Q will also converge or reach the optimal level.

$$\in_\theta \omega_\theta(s_t, a_t; \beta) = (r + \tau Q_\pi(s_{t+1}, a_{t+1}; w) - Q_\pi(s_t, a_t; w)) \in_\theta \log \omega_\theta(s_t, a_t; \beta) \quad (4)$$

$$loss(a_{t+1}; W) = \frac{1}{2} \lVert \hat{y}_t + \tau Q_\pi(S_{t+1}, a_{t+1}; w) - Q_\pi(S_t, a_t; w) \rVert^2 \quad (5)$$

4 Experiment

4.1 F-Model

In order to test the approximation ability of the F model, we input the training and valid Boston housing dataset into the F-model, and set several integers: 3, 10, 25, 50, 60, 100 as True labels to train the F-model. It can be observed from Fig. 3 and Fig. 4 that the F-model converges in all cases. Subsequently, input valid and training datasets into well-trained F-model in all cases for prediction. We can observe that the trained F-model has a good approximation ability (Fig. 4 and 5). Therefore, if we can obtain the best number of ANN layers and regard them as True labels, then use the training dataset and True labels to train the F-model, after training, when input similar dataset into the F-model, plausible predictions will be obtained. In addition, we normalized the training data to the range from 0 to 1, and trained the F-model as before, we found that the approximation performance of the F-model is better than before (Fig. 5, Fig. 6 and Table 6). Although those experiments have proved that our model has good approximation ability, we pay more attention to the generalization ability of F model, because our original intention is to let ANN know what they want to do, so F model should be able to approximate on two or more than two datasets.

Take the Iris and Boston housing datasets as input in turn, and set the True labels to 3, 10, 25, 50, 60, 100 in turn, as shown in Fig. 7, Fig. 8, and Table 7. The F-model can converge on both these two datasets at the same time, which indicates that the F-model has a generalization ability. Significantly, the generalization ability of the F-model is of vital importance, because the feature of the training dataset is not directly related to the RL stage, so the generalization ability of DSOL is mainly reflected in the F-model.

4.2 RL-Stage

First of all, let's restate the constraints of the ANN parameters. We set the initial layer number of ANN to 5 (Here we didn't use F-model to initialize the number of ANN layers because the F-model has not been trained. In the final test process of the experiment, we will use the trained F-model to initialize the layer number of ANN), and in the subsequent training process, the number of ANN layers should not exceed 15. In addition, we use the Golort Uniform method to initialize the weights of the ANN and set the bias of the initialized ANN layer to 0. In addition, except for the last and first layers of ANN, each Dense layer is followed by a Dropout layer with a probability of 0.5.

In the training process of the RL model, there are a total of 70 episodes, and in each episode, the RL model needs to optimize the ANN for 20 iterations. At the same time, we did not set the states of early termination for RL, because we have set the maximum number of ANN layers. According to related theories and some attempts, under the conditions we set, the performance of ANN increases with the number of layers. However, if we terminate the training of the model when the number of ANN layers reaches the maximum, we cannot prove that the RL model can stably set the ANN layer to the maximum in the

subsequent training process, which means that the model may not have collected all states, so we should train the model as many times as possible without excessive interference. The experimental results are shown in Fig. 9, Fig. 10, and Fig. 11. According to Fig. 10 and Fig. 11, we can observe that the loss function and accuracy of the ANN gradually converge with the training process. According to Fig. 9, Fig. 10, and Fig. 11, it can be observed that the performance of ANN is positively correlated with the increase in the number of ANN layers, and the well-trained RL model tends to add more Dense layers to the ANN, which is consistent with our expectations.

As mentioned in the methods section, we used the policy gradient ascent method to update the parameters of policy function ω and used TD error to optimize the parameters of the value function Q . Thus, both of them should converge after several episodes. Fortunately, the experimental results are consistent with the theory. As shown in Fig. 12, we can observe that both Q and ω converge into 0, which indicates that the RL stage has reached the optimal state. Significantly, maximizing the reward value of RL is equal to minimizing the reward value of RL multiplied by -1 . Thus, in Fig. 12 the TD error converges to zero indicates the RL model has attended to the maximum reward level. Finally, we use the optimal number of ANN layers to train the F-model, and then get the trained DSOL. As shown in Fig. 13, well trained DSOL performs well on the training dataset and can quickly optimize the number of layers of ANN to the optimal state, so that the loss function of ANN is always maintained at the lowest level.

4.3 Validation

We input 200 valid samples of the Boston housing dataset into a well-trained DSOL. As can be observed from Fig. 14, at the beginning of ANN optimization, DSOL sets the number of ANN layers to 15, and then in the subsequent ANN training process, the number of ANN layers does not change. At the same time, the loss function and accuracy of the ANN don't fluctuate greatly. These results show that the neural network has reached the optimal level in the initial stage of training. Thus, the well-trained DSOL performs well on the valid Boston housing dataset.

5 Conclusion and Perspective

We propose DSOL for data science. The model initially realized ANN depth self-configuration and self-optimization, so that the best ANN can be obtained on a specific dataset. In the experiment, we use 100 Boston housing training dataset samples for training, and use 200 Boston housing valid dataset samples to test the trained DSOL. According to the discussion in the experimental section, DSOL performs well. Significantly, the generalization ability of DSOL is mainly reflected in the F-model. Therefore, an F-model with a certain generalization ability is equivalent to a DSOL with generalization ability. After testing the F-model, we found that the F-model can approximate well on the Iris and Boston housing

datasets. Therefore, our DSOL has a certain generalization ability. *However, DSOL is just an initial form. We hope to develop a self-optimized system to handle various datasets in the future. In order to achieve this, we will mainly focus on those things:*

1. How to make the system optimize more kinds of ANN hyperparameters. We have known many NAS algorithms, which have been proved to be able to find new and feasible neural architectures. But we want to develop a model that can design the ANN architectures while taking datasets into account. That is to say, it's just like when a person processes a task, he will make a plan based on his analysis of the task.
2. How to make this system capable of dealing with more complex tasks rather than only data science. At present, most NAS algorithms can work on image datasets. However, this is a challenge for our model, because the image contains more complex information, so it may need to be improved with more complex logic to work.

Acknowledgment. We would like to acknowledge the two-month funding from the School of Lifescience in Tsinghua university, and thank one of my best friend Chuanxu Zhao for assistance in writing, and the reviewers for their valuable suggestions. Finally, best wishes to everyone who had ever encouraged and helped me and everyone who are working hard for scientific research.

Appendix

Architecture of Depth Self-optimizing Learning

See Tables 3 and 4.

Table 1. F-model on Boston housing dataset.

Layer type	Shape	Param
Input	(1,100, 13, 1)	0
Conv2D	(1, 100, 13, 64)	1664
Maxpooling	(1, 50, 6, 64)	0
Conv2D	(1, 50, 6, 64)	102464
Maxpooling	(1, 25, 3, 64)	0
Flatten	(4800)	0
Dense	(1, 128)	614528
Dense	(1, 64)	8256
Dense	(1, 1)	65
Total params: 726,977		
Trainable params: 726,977		
Non-trainable params: 0		

Table 2. F-model trained on Iris dataset.

Layer type	Shape	Param
Input	(1,100, 13, 1)	0
Conv2D	(1, 100, 13, 64)	1664
Maxpooling	(1, 50, 6, 64)	0
Conv2D	(1, 50, 6, 64)	102464
Maxpooling	(1, 25, 3, 64)	0
Flatten	(1600)	0
Dense (Relu)	(1, 4800)	7684800
Dense (Relu)	(1, 128)	614528
Dense (Relu)	(1, 64)	8256
Dense (Relu)	(1, 1)	65
Total params: 8,411,777		
Trainable params: 8,411,777		
Non-trainable params: 0		

Table 3. Policy function π

Input	(1, 2)
Dense (Relu)	(1, 10)
Dense (Relu)	(1, 5)
Dense (Softmax)	(1, 2)
Total params: 97	
Trainable params: 97	
Non-trainable params: 0	

Table 4. Value function Q

Input	(1,2)
Dense(Relu)	(1,10)
Dense(Relu)	(1,5)
Dense	(1,2)
Total params: 97	
Trainable params: 97	
Non-trainable params: 0	

Experiments of F-Model

Table 5. Performance of F-model on Boston housing dataset. Take the Boston housing training dataset as input, and take 3, 10, 25, 50, 60, 100 as real labels to train F-model in turn. The prediction results are shown in this table. In the first column, these predictions are obtained by inputting the training dataset into the trained F-model. In the second column, these predictions are obtained by inputting the valid dataset into the trained F-model. In the last column, these values are the true labels configured by ourselves.

Train dataset	Valid dataset	True layer
2.897	5.592	3.0
9.886	10.145	10.0
25.218	22.807	25
49.928	50.005	50.0
60.159	58.391	60.0
100.466	104.244	100.0

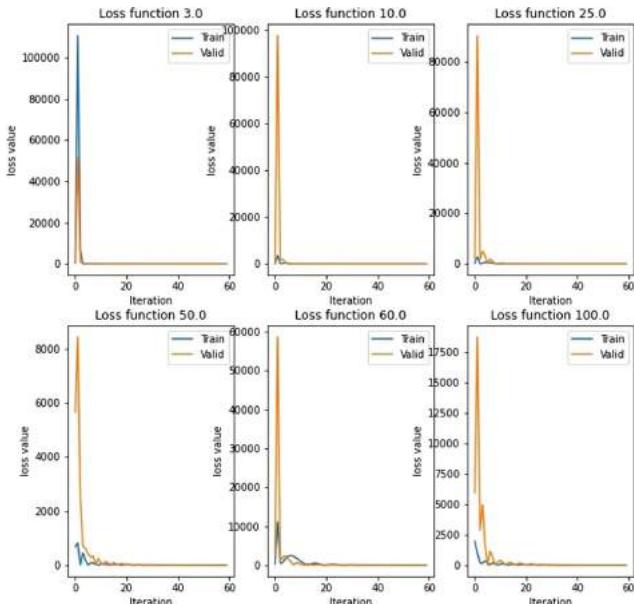


Fig. 3. Visualization of F-model loss function on Boston housing dataset. The Boston housing dataset is used as the input data, the true labels are set to integers: 3, 10, 25, 50, 60, 100 so as to establish the mapping from the input data to those integers by training F-model in turn. Then the loss function is plot in turn. From these functions, we could observe that the F-model converges in all cases.

Table 6. Performance of F-model on normalized Boston housing dataset. We normalize the Boston housing dataset to the range from 0 to 1, and still set integers: 3, 10, 25, 50, 60, 100 as true labels in turn. After training, we found that the approximation ability of F-model has been significantly improved. So we should normalize the training data before inputting it into F-model.

Train dataset	Valid dataset	True layer
2.944	2.936	3.0
9.932	10.139	10.0
24.962	25.426	25
49.766	50.902	50.0
59.958	60.906	60.0
98.728	100.162	100.0

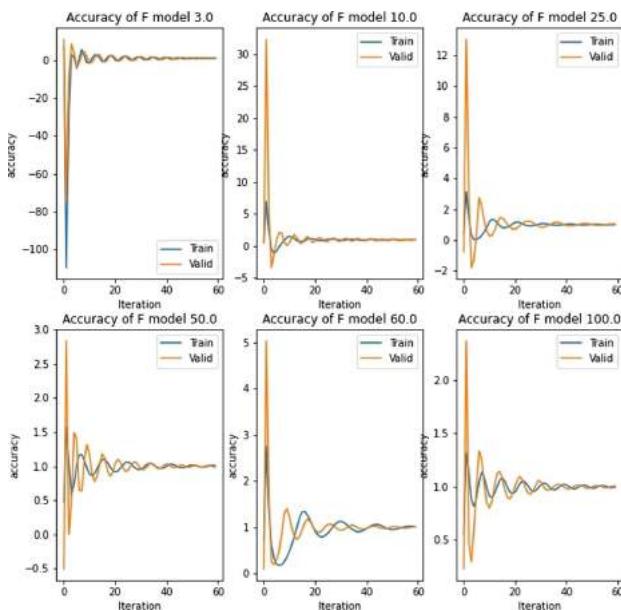


Fig. 4. Visualization of F-model accuracy on Boston housing dataset. Input the Boston housing valid dataset into the trained F-model, we obtain the predictions, then by dividing the predicted value with the true value, we get the precision shown in all cases. (True label as 3, 10, 25, 50, 60, 100 in turn)

Table 7. Generalization capability of F-model. In order to test the generalization ability of the F-model, we use Iris and Boston housing training datasets as input to train F-model in turn and then use Iris and Boston housing test training datasets as input to obtain the predictions shown in the table. The first column is the prediction obtained by inputting boston housing valid dataset into the F-model, the second column is the prediction obtained by inputting the Iris test dataset into the F-model, and the third column is the true label we set.

Boston housing	Iris	True layer
1.301	3.140	3.0
8.225	10.300	10.0
23.073	25.889	25
49.090	51.984	50.0
59.011	62.214	60.0
97.988	101.951	100.0

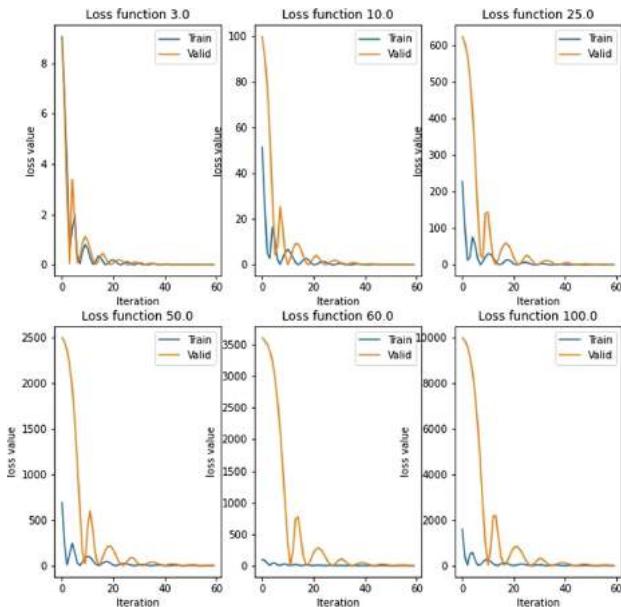


Fig. 5. Visualization of F-model loss function on normalized Boston housing dataset. After normalizing the Boston housing dataset, we observe an obvious enhancement of the approximation capability of F-model.

Experiments of RL-Stage

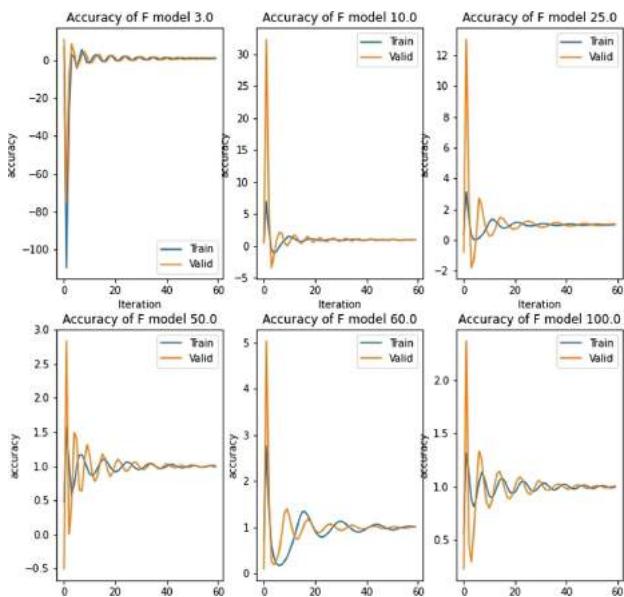


Fig. 6. Visualization of F-model accuracy on normalized Boston housing dataset. *Input the normalized Boston housing valid dataset into the trained F-model, we obtain the predictions, then by dividing the predicted value with the true value, we get the precision shown in all cases. (True label as 3, 10, 25, 50, 60, 100 in turn)*

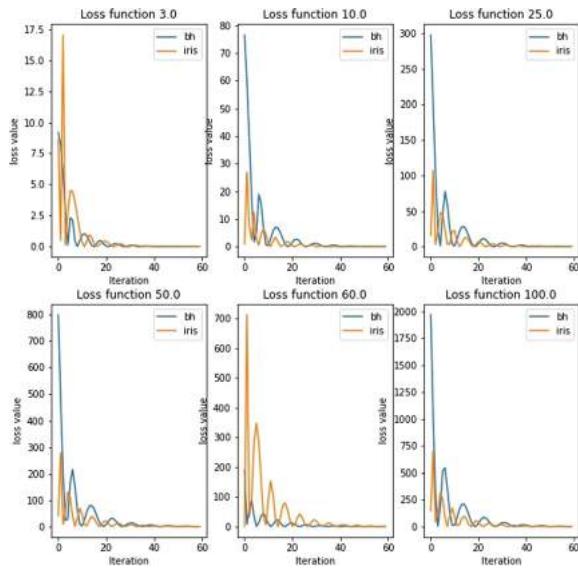


Fig. 7. Visualization of F-model loss function on normalized iris and Boston housing dataset. *In order to test the generalization capability of F-model, we use the Iris and Boston housing datasets as the input in turn and we can observe that F-model converge on both of Iris dataset.*

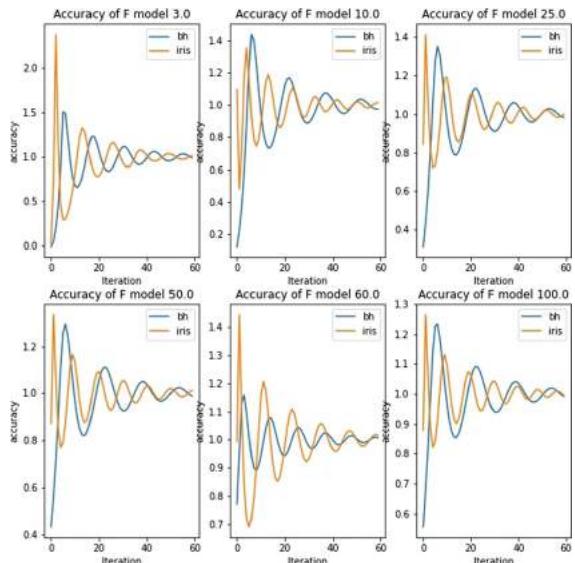


Fig. 8. Visualization of F-model accuracy on normalized Iris and Boston housing datasets. *Using Iris and Boston housing dataset to train F-model in turn, we can observe that F-model approximate well on Iris and Boston housing datasets.*

Validation of Trained DSOL

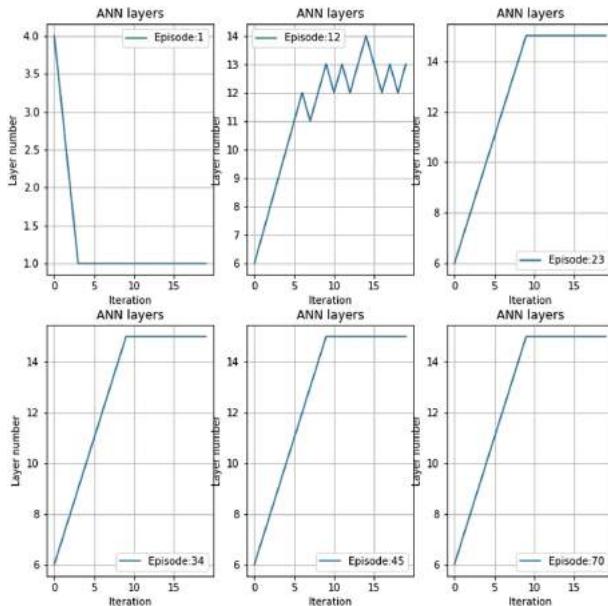


Fig. 9. Visualization of ANN layers. During the RL training process, we set up 70 episodes, and each episode contains 20 iterations. By plotting the changes in the number of ANN layers from episodes 1, 12, 23, 34, 45, and 70, we can intuitively observe that as the number of sets increases, the RL model tends to add more layers to the ANN.

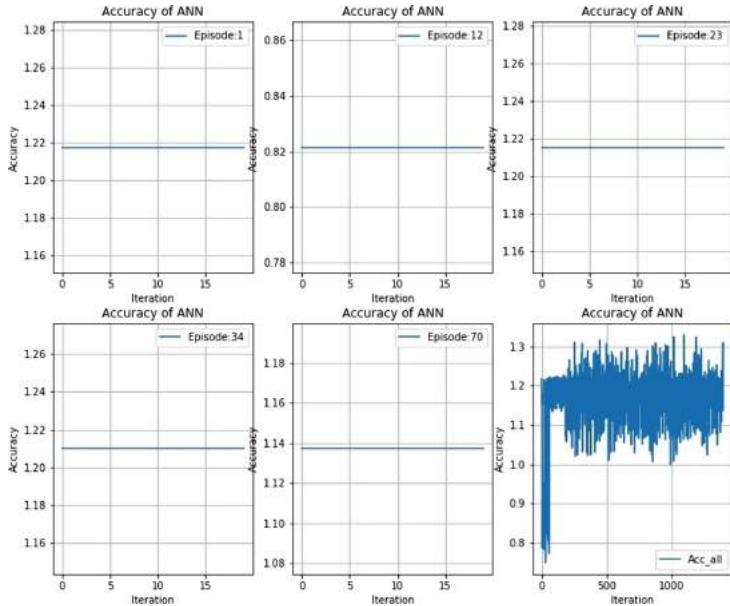


Fig. 10. Visualization of the accuracy of ANN. We sample sequential accuracy values from 1, 12, 23, 34, and 70 episodes in turn, and then plot these samples and total accuracy. In the last picture, we can roughly get the change trend of accuracy. During the ANN training process, its accuracy value gradually increases to about 1, and there is no abrupt change after that. It is worth noting that in order to obtain the accuracy value, we use the predicted value divided by the actual value, and then take the average value to represent the accuracy. This is only an approximate representation method, so the value obtained by this method may be greater than 1.

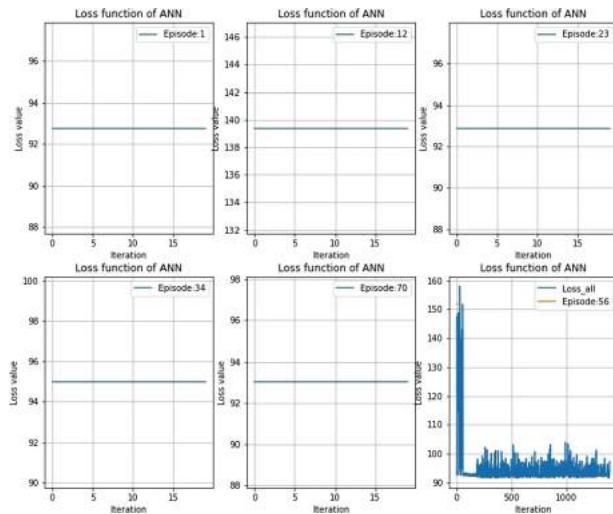


Fig. 11. Visualization of the loss function of ANN. The loss function of ANNs correspond to Fig. 10.

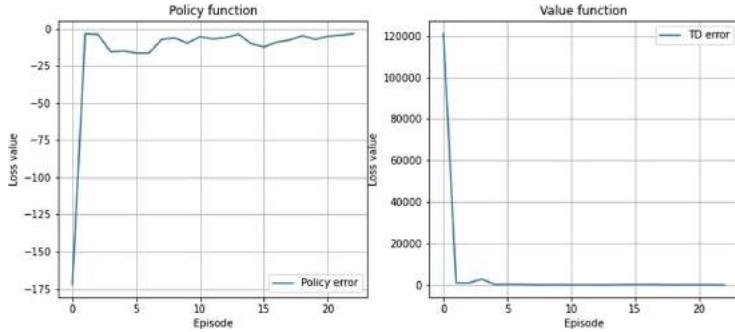


Fig. 12. As shown in the left figure the loss value of the policy function π converges to 0, which proves that π has attend to the most stable state π^* . significantly, during the experimental process, in order to maximize the TD value, we multiply TD by - 1 and reduce the processed TD value by the actor critic method. Therefore, as shown in the right figure when the processed TD value converges to 0, it also indicates that the benefit of the model reaches the maximum at this time.

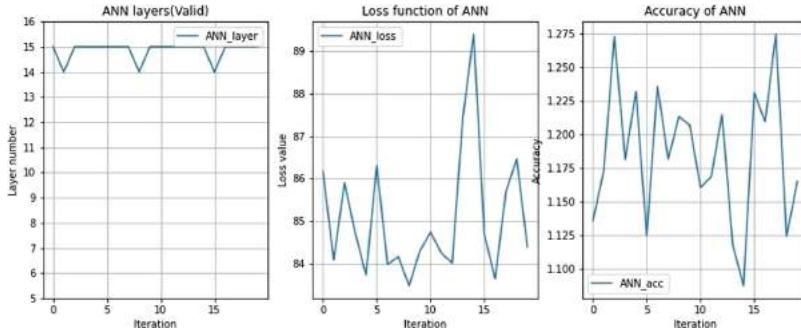


Fig. 13. Visualization of trained DSOL. Using trained DSOL to configure and optimize the ANN. We can observe that the number of ANN layers is optimized to the optimal level quickly by trained DSOL.

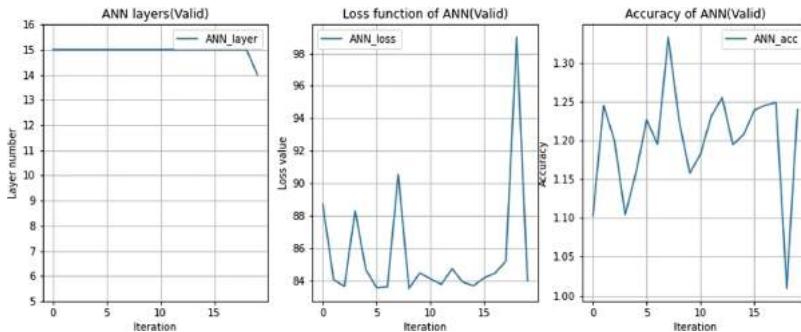


Fig. 14. Validation of trained DSOL. Input 200 samples of valid Boston housing dataset into the trained DSOL, record the performance of ANN and DSOL. We can observe that DSOL performs well on the valid Boston housing dataset.

References

1. Cybenko, G.: Approximation by superpositions of a sigmoidal function. *Math. Control Sig. Syst.* **2**(4), 303–314 (1989). <https://doi.org/10.1007/BF02551274>
2. Hornik, K., Stinchcombe, M., White, H., et al.: Multilayer feedforward networks are universal approximators. *Neural Netw.* **2**(5), 359–366 (1989)
3. Telgarsky, M.: Benefits of depth in neural networks. In: Conference on Learning Theory (2016)
4. Eldan, R., Shamir, O.: The power of depth for feedforward neural networks. In: Conference on Learning Theory (2016)
5. Lin, H.W., Tegmark, M., Rolnick, D.: Why does deep and cheap learning work so well? *J. Stat. Phys.* **168**(6), 1223–1247 (2017). <https://doi.org/10.1007/s10955-017-1836-5>
6. Poggio, T., Mhaskar, H., Rosasco, L., Miranda, B., Liao, Q.: Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *Int. J. Autom. Comput.* **14**(5), 503–519 (2017). <https://doi.org/10.1007/s11633-017-1054-2>
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press, Cambridge (2016)
8. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997). <https://doi.org/10.1109/4235.585893>
9. Elsken, T., Metzen J.H., Hutter, F.: Neural architecture search: a survey. arXiv preprint [arXiv:1808.05377](https://arxiv.org/abs/1808.05377) (2019)
10. Mnih, V., et al.: Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015)
11. Brown, N., Sandholm, T.: Libratus: the superhuman AI for no-limit poker. In: International Joint Conference on Artificial Intelligence (2017)
12. You, Y., Pan, X., Wang, Z., Lu, C.: Virtual to real reinforcement learning for autonomous driving. arXiv preprint [arXiv:1704.03952.2017](https://arxiv.org/abs/1704.03952) (2017)
13. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. arXiv preprint [arXiv:1611.01578](https://arxiv.org/abs/1611.01578) (2016)
14. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
15. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. arXiv preprint [arXiv:1707.07012](https://arxiv.org/abs/1707.07012) (2017)
16. Cai, H., Chen, T., Zhang, W., et al.: Reinforcement learning for architecture search by network transformation. arXiv preprint [arXiv:1707.04873](https://arxiv.org/abs/1707.04873) (2017)



Predicting Resource Usage in Edge Computing Infrastructures with CNN and a Hybrid Bayesian Particle Swarm Hyper-parameter Optimization Model

John Violas¹(✉), Tita Pagoulatou¹, Stylianos Tsanakas¹,
Konstantinos Tserpes², and Theodora Varvarigou¹

¹ National Technical University of Athens,

9, Heroon Polytechniou Str., 15773 Zografou, Greece

{violas, el14205, el09727}@mail.ntua.gr, dora@telecom.ntua.gr

² Harokopio University of Athens, 9, Omirou Str., 17778 Tavros, Greece
tserpes@hua.gr

Abstract. As the computational needs of edge infrastructures increased, efficient resource management becomes a necessity. An accurate prediction of future resource usage provides insight into dynamic task offloading, proactive auto-scaling, virtual machine migration, and workload balancing. In this paper we propose the use of multi-output one-dimensional convolutional neural networks as resource usage predictors. Convolutional neural networks can manipulate resource usage observations as time series data with the advantage of an adaptive window size selection. In addition, we propose an innovative hybrid hyper-parameter optimization method that combines particle swarm optimization and Bayesian optimization in order to conclude to a close to optimal convolutional neural network architecture. To validate the efficiency of our approach, we conducted experiments with an edge computing infrastructure. The evaluation results show that the proposed regression model achieves higher accuracy as compared to other machine learning meta-predictors and state of the are resource usage models.

Keywords: Resource usage · Edge computing · Convolutional neural network · Particle swarm optimization · Bayesian optimization

1 Introduction

The virtualization of resources has radicalized the world of computing especially during the last decade. Technology has gone a long way to spawning, migrating, replicating resources on demand, but there is a basic problem that persists: resource usage prediction [1, 2]. Accurately predicting the resources that a computational task will require in order to be successfully executed remains central in every cloud service pipeline. The advent of edge computing only serves to

magnifying the importance of a potential solution. Edge resources are typically of limited capacity and their intelligent allocation to computational tasks is of utmost importance.

The literature thrives with proposals employing methods such as benchmarks, statistical analysis, code analysis and other suchlike approaches. Lately, it is machine learning that has gained an increasing share in the proposals. Following this trend and leveraging the recent advances in the field of deep learning, we propose an approach to predict the computational load of a certain task in edge resources. The prediction methods that we propose is based entirely on historical monitoring data, thus decoupling the prediction from the application implementation and the resource characteristics.

The basic premise in our approach is that the problem of predicting the resource usage of edge nodes can be modeled as a time series regression problem. Sequential data might present similarities or dependencies with each other. Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) are able to recognise the similar data patterns and exploit them in order to address efficiently the problem of timely resource usage prediction.

To the best of our knowledge, in edge computing field does not exist a sufficient methodology for flexible adaptation to the temporary evolving resource usage metrics. In order to fill this gap we propose a method that executes a pipeline of tasks beginning with monitoring the current resource usage of edge nodes and leveraging historic data. Next, the monitoring observations are pre-possessed and fed to a multi-output regression CNN. The most challenging task in this workflow is to train a multi-output regression CNN. This challenge involves the selection of the optimal or close to optimal topology of the CNN, the estimation of the weights and biases of the CNN synapses, named parameters of the CNN and many other structural design decisions such as the activation functions and dropout rate. The structural design decisions are named hyper-parameters.

The training phase starts by seeking the optimal numerical hyper-parameters of the CNN model with the Particle Swarm Optimization (PSO) algorithm and the nominal hyper-parameters using the Bayesian Optimization (BO). The proposed hybrid-optimization method that combines PSO and BO for the estimation of close to optimal CNN model is titled Hybrid Bayesian Particle Swarm Hyper-parameter Optimization (HBPSHPO). HBPSHPO makes multiple iterations between proposing candidate CNN topologies and evaluate them based on historic data. As we will describe in the theoretical section of the paper and the experimental results, HBPSHPO makes a smart search in the CNN topologies converging quickly and eventually the trained multi-output regression CNN makes accurate predictions of the resource usage for the next timesteps.

In order to evaluate the proposed model we conducted real experiments with an edge computing infrastructure comprised of a cluster of Raspberry PI boards executing a Natural Language Processing (NLP) application. We deployed a python script on the edge devices in order to monitor their resource usage and constructed a training and evaluation dataset. Next, we conducted experiments

with the multi output regression CNN model which use the HBPSHPO model and we compared the results with other resource usage and machine learning state of the art prediction models available in the literature. The evaluation results show that the proposed model outperforms the prior work in terms of predictions accuracy i.e. root mean squared error and mean absolute error and confirm the applicability of our method.

The three major contributions of our research are:

- The proposal and analysis of multi-output regression one-dimensional CNN as resource usage predictors in edge computing infrastructures which outperforms previous time series and machine learning predictors using the convolutional function it applies an adaptive window size selection in the time series resource observations.
- The proposal of an innovative hybrid hyper-parameter optimization model that combines the PSO algorithm for numerical hyper-parameters with the BO for nominal hyper-parameters in order to gauge a close to optimal neural network topology.
- An experimental evaluation in an edge infrastructure of a multi-output one-dimensional CNN gauged by the proposed hybrid hyper-parameter optimization model and comparison with other state of the art machine learning meta-models and resource usage predictors. The outcomes show significant improvements of our proposed methodology in terms of accuracy.

The rest of the paper is structured as follows: Sect. 2 highlights the related work in resource usage prediction, the related machine learning and hyper-parameter optimisation techniques. Section 3 explains the functionality of resource usage prediction in the administration of edge computing infrastructures. Section 4 provides an analysis of the multi-output regression CNN and the HBPSHPO method. Section 4 describes the experimental setup and the evaluation results. Finally, Sect. 5 concludes the paper and suggest directions for future work.

2 Related Work

Resource usage prediction is often presented as a function of “workload prediction”, task implementation details and resource type. This is particularly the case with a body of literature where the authors assume the latter two as constants, attacking solely the problem of workload prediction. In these works the aim is often to identify workload patterns that appear across various cloud tasks’ execution. The key challenge is to identify the patterns [3], or workload models [4] and characterize the workloads based on whether they “include” them or not. Such models are very useful as benchmarks in performance modelling techniques [5]. However, the limitation of generating solutions for fixed application and resource types as well as the broad concept of workload and goals [6] pose strong constraints in the ability of such solutions to generalize.

Making predictions based on monitoring data seems to alleviate this problem providing the basis for application- & resource- independent workload prediction methods. On that basis, there is a growing body of literature attacking the problem using machine learning and statistical analysis approaches. For quite a long time, a large set of configurations have been tested, including simple perceptrons [7] and linear regression [6], ARIMA models [8] and deep learning with an emphasis on the reduction of the problem's dimensionality [9].

The common downside of these approaches is that a new model needs to be developed for every application-resource type pair. To tackle the latter problem and create more generic solutions, researchers have developed unsupervised learning approaches such as Principal Component Analysis (PCA) to identify resource usage patterns across different nodes [10].

The administration of edge computing infrastructures involve a set of real time and proactive scheduling strategies [11] to ensure the smooth operation of applications, guarantee the quality of services, minimize computational latency and optimize bandwidth allocation, energy consumption and total cost. The fluctuation of computation workload, the dynamicity of users behaviour and the heterogeneity of edge nodes urge the research community to propose intelligent and automated mechanisms. Deep learning methods have been widely used lately for the orchestration of edge and cloud infrastructures with being the core models of task offloading [12], resource management [13] and service migration [14] just to mention a few.

The tasks execution on cloud or edge computational resources mostly does not scale linearly with their input making linear resource usage models not sufficient and machine learning solutions a reasonable approach. Single-output regression feedforward neural networks [15] have been used for the resource usage prediction on a given task and historic data. Resource usage observations can be represented and analysed as time series data. The previous simple feedforward Neural Network model has the limitation that cannot leverage the sequential structure of information. Long Short-Term Memory (LSTM) [16] is a specific layer of RNN which leverage time series and has been used to model for a given workload the relationship between resource allocation and cpu and memory usage.

The LSTM model uses a fixed size observation window. If this window is short, then it cannot capture some changes in the resource usage trends. If the fixed size window is irrelevant large then an irrelevant large number of observations may yield inaccurate estimations. To address this limitation a deep learning-based adaptive window size selection [17] has been proposed for resource usage prediction which leverages different size trend periods. A slight different adaptive resource usage prediction model builds multiple data representations which are extracted by multiple window sizes and uses a set of classical machine learning models with random decision forests approach [18]. Random decision forests belong in the category of ensemble learners and meta-learners. Ensemble methods have also been combined with feature engineering techniques [19] in order to provide accurate resource usage predictions in a meta-learning fashion.

Taking into consideration the aforementioned recent developments in the edge computing resources management, there is a great interest of: (i) how to use deep learning models as accurate regression predictors; (ii) to leverage the time series of resource usage observations in an adaptive way; (iii) to use a meta-learning approach to search for an optimal model. In order to satisfy these three requirements we propose a deep learning model with CNN layers and an innovative hybrid hyper-parameter optimization model as a meta-learning algorithm.

Our proposed model, to the best of our knowledge, is differentiated and surpasses the related work for resource usage predictors in edge computing infrastructures for two reasons. First, we use multi-output regression CNN to predict the usage of multiple resources in a unified way and leverage the time series observations in an adaptive window size. Second, we propose an innovative hybrid meta-learning model that combines PSO and BO in order to estimate the close to optimal nominal and numerical deep learning hyper-parameters. The analysis and usage of the CNN layers, PSO and BO are discussed in separate sections in this paper.

3 A Convolutional Neural Network Approach

The prediction of edge nodes resource usage in an edge computing infrastructure can involve many features from different devices. Each data feature presents a sequential structure and the resource usage features have intrinsic dependencies the one with the other. In order to leverage the specific data characteristics monitored by edge nodes and the target values of the next time step resource usage we propose the use of multi-output regression one dimensional CNN. We use multi-output regression because we predict for each device the usage of multiple outputs such as cpu, memory, bandwidth usage as scalar values and it is one dimensional convolutional network because the neural network captures the sequential information.

Neural network models are full of parameters and hyper-parameters. Parameters include the weights and biases of the connections between the neurons. Parameters are calculated using a learning algorithm such as gradient descent and its variants. Hyper-parameters determine the network structure and how this network is trained. The network structure includes the number of layers and the number of neurons for each layer. The decisions regarding how the network is trained may involve the selection of learning algorithm, the learning rate and the batch size. The selection of the hyper-parameters is not trivial and a trial and error approach may lead us to large amount of computational expensive trials with great uncertainty for the selected hyper-parameters. In order to make a smart search in the hyper-parameter space we will use HBPSHPO model that estimates the nominal hyper-parameters such as the activation function with a Bayesian optimization approach and the numerical hyper-parameters such as the number of layers with an approach based on the Particle Swarm Optimization.

The steps of the proposed model for the resource usage predictions is depicted in Fig. 1 as a pipeline. In the beginning we monitor the edge devices and build

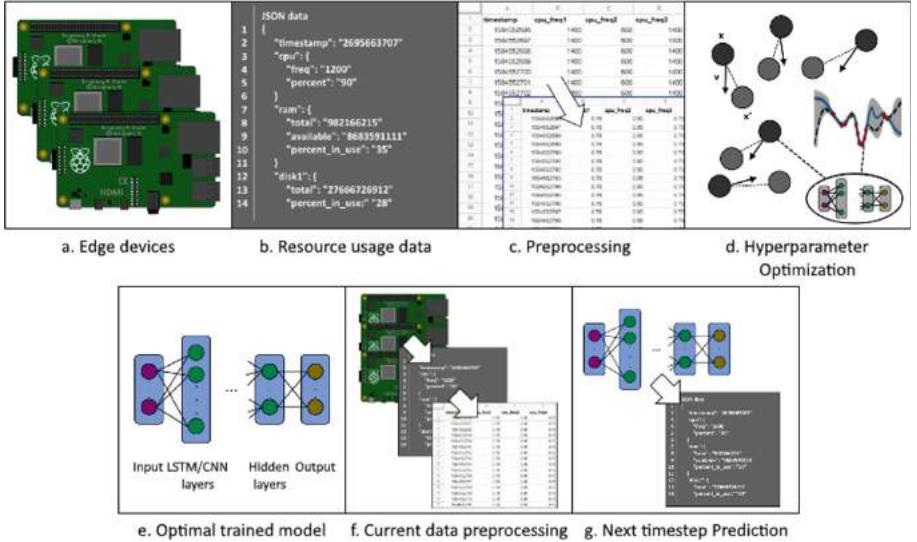


Fig. 1. Pipeline of training and inference with multi-output regression CNN with the HBPSHPO

a training dataset. This dataset is pre-processed including cleaning and scaling the data values. Next, the dataset is fed to a set of candidate neural networks with variable hyper-parameters defined as the searching space of the particles of PSO and BO. The particles of PSO make multiple iterations searching in the hypothesis space in combination with the parameters of the BO method. The hyper-parameter optimization process concludes to a close to optimal trained neural network. This neural network provides next time-step predictions based on the current and previous resource usage observations. The quality of training data is the main limitation of the proposed approach. In order to provide accurate predictions the input observations should have the same statistical properties with observations of the training data. This characteristic exists in all data driven methods.

In the following subsections we will describe the key parts of the multi-output regression CNN model including the regularization and learning process, the LSTM which also examined as a potential approach but eventually they had lower accuracy than CNN and the HBPSHPO model.

3.1 Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of feed forward neural network which is suitable for processing data which have a grid structure. Its architecture usually consists of one or more convolutional layers with subsampling stages and one or more fully connected layers as it happens in common multi-layer neural networks. CNNs use an optimizable feature extractor named kernel, which

enable CNNs to easily memorize spatial order of features. The latter helps CNNs to perform an important accuracy and efficiency. A convolution is performed between the data matrix and the kernel matrix, with the convolution window moving n elements for each multiplication, n parameter called strides. After the convolutional layer, a common practice is adding a pooling layer, either a max or an average one. The purpose of the pooling layer is to reduce dimensionality and thus the computational power required to process the data, as well as to suppress the noise. Following the convolution and pooling layers, the data is flattened, so it can be used as input to a fully connected feed forward network, which can help learn non linear patterns and features.

In the case of resource usage prediction, as we can see in Fig. 2, CNNs featuring one-dimensional convolutional layers with max pooling layers were used. The input data comprises of a time-series that show the percentage usage of every resource available to our system, such as cpu, ram, bandwidth. The notion is that one-dimensional convolutional layers will help find patterns that refer to the usage of our resources, performing the convolution on the time dimension. Particularly, after splitting our dataset timeseries into its train and test sets, we transform it into samples of shape (n, features). This is fed into our network, which performs one dimensional convolutions on each sample, the idea being that resource usage of a specific timestep has a much higher correlation to more recent measurements of the resource usage compared to earlier ones. This way the network is trying to predict the feature values of the next timestep using the last n values of the dataset.

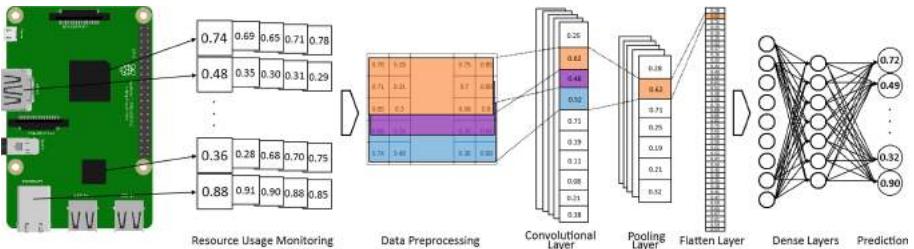


Fig. 2. Convolution neural networks for resource usage prediction

3.2 Long Short-Term Memory

Long short-term memory (LSTM) is a type of RNN which contains memory blocks in the recurrent hidden layers. Specifically, a LSTM unit includes the cell, the input gate, the output gate and the forget gate, which ensure the storage and access in previous data, even after the passage of large temporal periods. Thanks to its ability capture long-term temporal dependencies, LSTM is suitable for time series problems. It is also capable of tackling two common problems of RNNs: the short-term memory and the vanishing gradient problem.

The main drawback of LSTM for the resource usage prediction is that use a fixed size observation windows which can not being adaptively adjusted to capture local trends in the most recent observations or distant trends in the older data sequences [17].

3.3 Regularization

A neural network should be capable to generalize to new data inputs never seen before. A regularization practice often used is the dropout technique, in which a percentage of the recurrent connections are excluded from activation during each learning iteration. Early stopping is also a regularization strategy in which we evaluate the model in each training iteration and when the accuracy is decreased in the testing dataset, the training stops.

3.4 Learning Process

The parameters of a neural network can be optimized by minimizing an objective function. The objective function expresses the difference between the predicted output and the actual output such as Mean Absolute Error L1 and Mean Squared Error L2. Objective functions can also be combined with penalization weight techniques in order to regularize the model. Based on the existing literature the most widely used optimization techniques for CNN-RNN and LSTM-RNN are the Root Mean Square Propagation (RMSProp) and the Adaptive Moment Estimation (Adam) [20]. RMSProp and Adam are both Stochastic Gradient Descent approaches with an adaptive learning rate. RMSProp uses the Momentum approach, in which the gradient in every iteration is the sum of the current gradient and the previous gradients, which results in restricting the oscillation. Adam, similarly uses the Momentum technique, but it also finds an invariant direction of slope in contrast to the other oscillating directions, with the navigation through saddle points.

3.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) [22] is an evolutionary algorithm that aims to locate the optimal hyper-parameters and minimum value of the objective function. Evolutionary algorithms are metaheuristic optimization algorithms that simulate mechanisms inspired by nature. PSO attempts to mimic the behaviour of a flock of birds or a shoal of fish. In other words, the “birds” fly through a N-dimensional search space, where each dimension represents a single hyper-parameter and every position in the search space corresponds to a hyper-parameter set from which we can extract the objective function’s loss value via a neural network training and evaluating procedure. The main elements of the PSO algorithm are depicted in Fig. 3 and summarized as follows.

Firstly, PSO requires an objective function to minimize. The objective function in deep neural networks usually includes the training process of the network

which returns the loss value of the evaluation method. Secondly, we should define the number of searching agents - particles in the swarm (swarm size), which represent possible solutions of the problem, and the maximum number of iterations that the swarm is going to run for. The agents explore the search space in every iteration in order to find the minimum value of the objective function updating their position x and velocity v in each iteration. During the search procedure, each agent saves its personal best value p_i and the hyper-parameter combination that produced value. The agents can interact with one another and thus, in case of an agent's personal best value being the best among all the other agents up until that point, this agent informs the rest of the agents and save its value as the global best value p_g . As in any hyper-parameter optimization problem, there is no objectively best swarm size value one should choose since the optimal swarm size depends on the specific problem. However, using a large number of particles may lead to slow convergence with no noticeable improvement of the performance.

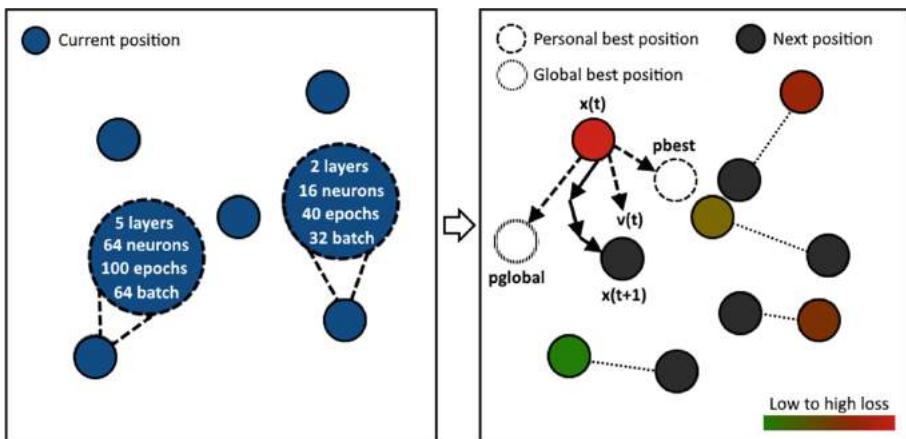


Fig. 3. Particles make a smart search

Thirdly, we choose the PSO hyper-parameters σ , ϕ_p and ϕ_g which represent the particle velocity scaling factor, the scaling factor to search away from each particle's best known position and the scaling factor to search away from the swarm's best known position respectively. These hyper-parameters are used to confront two important concepts of every optimization algorithm: exploration and exploitation. The former refers to the ability of the particles to search away from their current position, whereas the latter represents the ability of the particles to search the surrounding area of their current position. Specifically, σ also called inertia is a positive constant, typically between 0.4 and 0.9, that is reduced as the algorithm progresses. The σ defines the exploration attribute by specifying the influence the agents' previous velocity has in its next position. The parameters ϕ_p , ϕ_g are non-negative coefficients that define the acceleration

weight towards the personal and global best solution respectively. Decreasing ϕ_p the exploitation of the search space is also reduced, whereas as we decrease ϕ_g the exploration is decreased. Balance between exploration and exploitation is achieved by tuning the σ , ϕ_p , ϕ_g accordingly. The PSO algorithm is summed up as follows. Firstly, we initialize the positions and velocities of the particles randomly. Then, we execute the training procedure which produces the loss value for each particle, and update the personal and global best results, if necessary. Afterwards, we update the velocity and position of each particle for the next timestep according to the Eqs. 1 and 2.

$$v_i(t+1) = \sigma \times v_i(t) + \phi_p \times \rho_1 \times (p_i - x_i(t)) + \phi_g \times \rho_2 \times (p_g - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

where ρ_1 and ρ_2 are random numbers in range [0, 1]. After the above computations are completed, we proceed with the training procedure for the next iteration. Once a termination criterion is satisfied, such as reaching the maximum number of iterations, the algorithm terminates and p_g is the best result that PSO identified.

3.6 Bayesian Optimization

Bayesian Optimization (BO) [23] is a statistical method used for optimization of black-box functions - functions that their internal expressions and mechanisms are not known. Thus, BO is suitable for optimization problems with specific limitations that a common optimization algorithm could not tackle. One such limitation is that the function's derivatives are unknown, and consequently the optimizer does not have evidence for the direction of the function. Secondly, the optimizable objective function may be expensive to calculate, therefore we should narrow down the samples of inputs that the function will compute eventually.

BO uses two different types of functions in order to locate the global optimum, the surrogate and the acquisition function. A surrogate function attempts to approximate the objective function by sampling inputs, which enables us to assume more confidently where optimal points are more likely to be. A widely used surrogate function is the Gaussian Process (GP), which provides a prior over functions and is able to assimilate prior beliefs about the objective function.

Acquisition function is responsible for suggesting the next sampling points that should be examined, balancing at the same time exploration and exploitation. In BO, exploration describes the possibility to search points which present high uncertainty on being an optimum point, whereas exploitation defines the possibility of surrogate function to estimate points located in high-objective estimated regions. An acquisition function that suggests exploration and without exploitation, leads to the exploration of the whole search space, even if a good point near the global optimum is found, whereas exploitation without exploration causes the exploration of a limited area of the first optimal point found.

The BO algorithm is described by the following steps. Firstly, we initialize a GP surrogate function prior on a function f , estimate n_0 points where the

current prior distribution is optimized, and calculate these n_0 points in the objective function. Afterwards, we update the prior distribution using the updated available data to extract a posterior, which will become the prior in the next iteration. We repeat the above procedure, until the maximum number of iterations is reached, and finally, we explicate the final GP distribution in order to locate the global optimum.

3.7 Hybrid Bayesian Particle Swarm Hyper-parameter Optimization Model

HBPSHPO used both PSO and BO with GPs for the hyper-parameter optimization. The former was responsible for the numerical hyper-parameters, such as units and layers, and the latter for the nominal, such as activation functions and optimizers. The reason that separation occurred is due to PSO's inability to manage nominal hyper-parameters easily, whereas BO is able to handle both numerical and nominal hyper-parameters. The algorithm of HBPSHPO is given by Algorithm 1.

In the HBPSHPO, the numerical hyper-parameters are defined by PSO and include units, layers, dropout rate, lookback, learning rate, epochs, batch size and number of LSTM/CNN layers. In order to have a complete deep neural network, the building of the network requires optimizers and activation functions for each layer of the network. This task is implemented by the BO which functions inside the PSO algorithm and searches the optimal nominal hyper-parameter set given a numerical hyper-parameter set from PSO. Once the optimal nominal hyper-parameter set is found, the deep network gets trained and returns the loss value from the evaluation method.

4 Experimental Results and Discussion

4.1 Experimental Setup

The multi-output regression CNN with the HBPSHPO model is implemented in Python 3 using the frameworks NumPy, pandas, statistics, Scikit-learn, TensorFlow 2, SciPy, PySwarms and Scikit-Optimize. The environment we used is the Jupyter notebook of the Google Colaboratory. The experiments' source code is available for any kind of reproduction and re-examination at the second author's GitHub repository [21].

The dataset constructed by a monitoring tool implemented in Python 3 that uses the libraries psutil [24] and GPUUtil [25]. We monitored the real time usage of CPU, RAM, Disk I/O and Bandwidth in one second time interval. The edge nodes were Raspberry Pi3 with a 64-bit quad-core ARM Cortex-A53 at 1.4 GHz with Raspbian operating system which is a version of Debian Linux. The deployed application was a natural language processing text classification. We decided to make the text classification on an edge computing environment, locally, close to the text owners and not in Cloud computing infrastructures for

Algorithm 1. Hybrid Bayesian & PSO Optimization Algorithm

Step 1: Initialization of PSO

For each particle $i=1,\dots,M$ do

- i) Initialize randomly the position of each particle $x_i(0)$ between the lower and upper bounds of the search space
- ii) Initialize randomly the velocity of each particle $v_i(0)$ within the velocity range
- iii) Initialize the particle's best position to its initial position: $p_i(0) = x_i(0)$
- iv) Initialize the swarm's best position to the minimum value among all the initial positions: $p_g(0) = \text{argmin}(x_i(0))$

Step 2: Repeat until a termination criterion is satisfied**Step 2a:** Update of PSO's variables

For each particle $i = 1, \dots, M$ do

- i) Update particle's velocity according to the formula:

$$v_i(t+1) = \alpha \times v_i(t) + \lambda_p \times \rho_1 \times (p_i - x_i(t)) + \lambda_g \times \rho_2 \times (p_g - x_i(t))$$
- ii) Update particle's velocity according to the formula:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

Step 2b: Bayesian Optimization with GP

- i) Apply a Gaussian Process prior on f
- ii) Observe f at n_0 points according to an initial experimental design
- iii) Initialize $n = n_0$
- iv) Repeat while $n \leq N$
 - a) Update the posterior probability distribution on f using all available data
 - b) Let x_n be a maximizer of the acquisition function over x .
 - c) Observe $y_n = f(x_n, x_i(t+1), v_i(t+1))$.
 - d) $n \leftarrow (n + 1)$

Step 2c: Update of PSO best positions

- i) If $y_n < p_i(t)$, then
 - a) Update i_{th} particle's best position: $p_i = x_i(t)$
 - b) If $y_n < p_g(t)$, then update swarm's best position: $p_g = x_i(t)$
- ii) $t \leftarrow (t + 1)$

Step 3: Output p_g which is the best found solution

privacy issues. The reason for this choice is that the text owners didn't agree their texts to be transferred and processed in remote servers. In order to control the application remotely and take the resource usage datasets we used the SSH protocol, but we didn't have the privileges to access the processed texts.

The proposed model is compared to four state of the art predictors, AUCROP [26], XGBoost [27], Auto-sklearn [28] and GA-LSTM [29]. AUCROP is the abbreviation of Application and User Context Resource Predictor which is a meta-model predictor which utilizes common machine learning algorithms for resource usage predictions. Auto-sklearn is a general-purpose automated state of the art machine learning meta model used for data pre-processing, regression and hyper-parameter tuning through the Bayesian Optimization algorithm. Its efficiency is attributed to its possibility of saving previous optimization runs which provides the opportunity for the training of data with previous saved settings. Auto-

sklearn has won the prestigious ChaLearn AutoML challenge, it is popular in research papers and it is considered as one of the best AutoML frameworks by the data scientist community.

XGBoost is an open-source software library which provides a gradient boosting framework and is often used for the Gradient boosted decision trees. Its availability in several programming languages and its integration in Python's libraries, such as Scikit-learn, have contributed to its popularity. XGBoost has been used a lot in Kaggle and KDD Cup winning solutions. The GA-LSTM model is our previous resource usage prediction model in edge devices that leverage genetic algorithms of the estimation of hyper-parameters and exploit, feedforward, RNN and specifically LSTM layers.

The proposed model used the out-of-sample evaluation technique which preserves the sequence of the observations. Therefore, we did not apply any shuffling method in the context of time-series applications. We split the dataset into two parts, the training sequence which was the 66% of the observations and the testing sequence which was the rest 34% of the data. The proposed model was evaluated with the evaluation metrics Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). We recorded the training time and the inference times for a single prediction and for a batch of 100 predictions.

The HBPSHPO which includes the PSO and BO algorithms realized a smart search in the hypothesis space in order to identify the optimal CNN Multi-Output Regression model. In our experiments with HBPSHPO, we also included hyper-parameters for GA-LSTM and feedforward layers. The experimental results of our proposed model, AUCROP, XGBoost, Auto-sklearn and Genetic Algorithm with GA-LSTM are summarized in the Table 1.

Table 1. Single-output & multi-output evaluation

Method	RMSE	MAE	CPU-1 (%)		RAM-1 (%)		Train. time	Infer. time	
			RMSE	MAE	RMSE	MAE		Single	Batch
PSO-CNN	<u>0.0631</u>	0.0254	<u>15.932</u>	12.898	<u>1.416</u>	0.587	1692	0.036	0.039
PSO-LSTM	0.0661	0.0276	16.088	<u>12.691</u>	1.479	0.613	1042	0.034	0.035
AUCROP	0.0814	0.0414	17.235	14.009	2.480	1.482	522	<u>0.004</u>	<u>0.011</u>
XGBoost	0.1139	0.0599	16.457	13.569	1.515	<u>0.472</u>	<u>181</u>	0.060	0.010
Auto-sklearn	0.1055	<u>0.0243</u>	52.659	17.856	1.546	0.526	1338	0.263	0.572
GA-LSTM	0.0674	0.0338	16.099	12.838	1.746	0.917	574	0.020	0.024

4.2 Results and Discussion

Our experimental study is divided into four main parts. At first, we examine the accuracy of models predictions in terms of average RMSE and MAE for all the monitored resources. Next, we provide for the worst predicted device, from the

edge infrastructure, the predictions accuracy of CPU and Memory. Afterwards, we examine the training time, the inference time for a single prediction and the inference time for a batch of 100 predictions. Finally we examine the convergence of the HBPSHPO method.

In the Table 1 we can see that the deep learning models i.e. PSO-CNN, PSO-LSTM and GA-LSTM have better accuracy than the other machine learning meta-models in average resource usage predictions, except the case of the single resource usage of RAM, where XGBoost achieves a better MAE performance. The PSO-CNN performs well on the average MAE with a 0.0255 value. The RMSE which is a metric that penalize large errors is greater than MAE declaring that few specific predictions have a greater prediction error. The CPU and RAM MAE of the device with the greater error is 12.691% and 0.613% respectively. We derive two conclusions from these values. Firstly, we can see the prediction limits of our resource usage model and secondly we understand that predicting RAM usage is far more accurate than CPU. In order to reason the second conclusion, we examined the resource usage dataset and we found that the CPU usage had more intensive fluctuations than the RAM.

Regarding the training time in the Table 1 we can see that it is not the minimum for the deep learning models. The reasons are twofold. First, the training of the DL models is a much more computational heavy process due to the back propagation and the great number of parameters than the training of traditional machine learning models. Second, the hyper-parameters of the deep learning models are much more than the hyper-parameters of the classical machine learning models. The reason that HBPSHPO has longer training time and predictions accuracy than other meta-models and specifically the genetic algorithm approach is that the HBPSHPO has a much greater granularity in the numerical hyper-parameters. The PSO hyper-parameters have numerical ranges with many candidate positions defined by a minimum stepsize. This provides the flexibility to approach closer the optimal solutions increasing the predictions accuracy but making more searching movements. The single and batch inference time in all cases is much shorter than 600 ms and we do not consider it noteworthy to improve it further. All the times in the Table 1 are expressed in seconds.

The HBPSHPO method made a smart search in the RNN-LSTM, CNN and simple feedforward neural network topologies and concluded that a neural network that begins with CNN and max-pooling layers, followed by one flatten layer, one feedforward, one dropout and one last dense layer has the best accuracy. The HBPSHPO selected Adam optimizer. The fact that Adam has also been proposed as the best optimizer in multiple CNN models [20] reassures the applicability of HBPSHPO for optimal hyper-parameter selection. Regarding the overall comparison of the hyper-parameter optimizations methods, we have seen that the HBPSHPO method surpasses the genetic algorithm. Specifically, we run several experiments with the same set of hyper-parameters between HBPSHPO and GA and the HBPSHPO had always better results.

Table 2. Numerical hyper parameters optimized with HBPSHPO

Value	Units	Layers	Lookback	Dropout	Learn. rate	Epochs	Batch size	Num CNN or LSTM
Min	1	1	1	0	0.001	20	32	1
Max	128	5	5	0.5	0.2	200	1024	2

Table 3. Nominal hyper parameters optimized with HBPSHPO

Optimizers	RMSprop	Adam	SGD	Adagrad	Adadelta	Adamax	Nadam
Activation function	tanh	linear		sigmoid		relu	

The hypothesis space in which the particles of the PSO are moved is provided in Table 2. The range of each hyper-parameter is provided by the rows Min and Max. We made some experiments with different configurations of the PSO and BO and concluded to set the swarm size of the PSO to 15 and the minimum change of swarm’s best objective value before the search terminate to 0.0001. The candidate nominal values of the BO part of the HBPSHPO hyper-parameters are provided in the Table 3 and we set the expected improvement as the acquisition function. The units hyperparameter defines the number of convolutional filters, LSTM units, as well as the number of dense layer neurons provided that this number is halved after each layer. Layers refers to the number of dense layers our network has, and lookback is the number of timesteps each training sample has. Moreover, the number of convolutional/pooling or lstm layers is treated as a hyperparameter, as well as the epochs, Learning Rate and Batch size of training. Finally, Dropout is added after each dense layer to avoid overfitting, with its hyperparameter defining its effect frequency. The hyperparameters that get handled by the bayesian analysis model refer to the activation function used by our network and the optimizer that is chosen for the training process.

Another desirable property of an optimization algorithm is the ability to converge. PSO is a meta heuristic approach based on systematic progression of search and evaluation of candidate solutions [30]. Bayesian optimisation based on Gaussian process provably converge to global optimum assuming that the kernel is known in advance. But this does not happen in most of the cases [31] and specifically in the nominal hyper-parameters of the neural networks. Consequently, the HBPSHPO convergence has not mathematical proof. However, the convergence can be experimentally measured based on the best and median error of the candidate solutions for each generation. Best error expresses the best loss value of the candidate neural networks (particles) in one generation of the swarm. Median error expresses the median loss value of all the candidate neural networks (particles) in one generation of the swarm. In all the experiments, the neural network topologies converged quickly and we didn’t have significant improvements in error rates after the first ten generations.

In the Fig. 4 we can see two cases where in the left graph we tested HBP-SHPO with only LSTM and feedforward layers and in the right graph we tested CNN, LSTM and feedforward layers. The HBP-SHPO model has been designed, to always survive the best particles from generation to generation and have a strong affect to the other particles of the swarm. So the the best error will always be improved or stay constant. Regarding the median error we see that the best particles affects the rest particles of the swarm and all together are moved towards most accurate ANN topologies. In both cases we see that we have a very fast convergence and after four to nine generation we can find a close to optimal neural network topology.

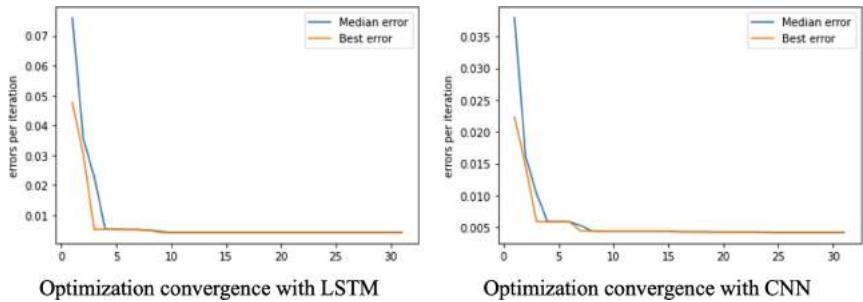


Fig. 4. Convergence of the hybrid Bayesian particle swarm hyper-parameter optimization model.

5 Conclusion

In this paper, we discussed how the prediction of resource usage can provide useful insight for many edge computing orchestration functionalities such as task offloading, workload balancing, proactive auto-scaling and fault tolerance. Resources usage is difficult to be predicted cause of the dynamicity and heterogeneity of the tasks and edge nodes. To address these challenges, we proposed a deep learning model with CNN layers that leverages the sequence of data observations and a hyper-parameter optimization method that combines BO and PSO in order to make a smart search in the space of nominal and numerical hyper-parameters of the deep learning models. The experimental evaluation showed that our proposed model provides very good predictions and surpasses other machine learning meta-models and resource usage estimators.

The future directions of this work is to implement a task offloading mechanism that uses the predictions of CNN HBP-SHPO model in order to calculate which processing edge nodes will have adequate resources available in the next time periods and satisfy the quality of service requirements. We plan to evaluate the performance of the task offloading model with an edge simulator like EdgeCloudSim, FogNetSim++ or iFogSim and afterwards to make experiments in real edge infrastructures.

Acknowledgments. This work is part of the ACCORDION project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871793.

References

1. Rahamanian, A.A., Ghobaei-Arani, M., Tofiqhy, S.: A learning automata-based ensemble resource usage prediction algorithm for cloud computing environment. *Futur. Gener. Comput. Syst.* **79**, 54–71 (2018). <https://doi.org/10.1016/j.future.2017.09.049>
2. Amiri, M., Mohammad-Khanli, L.: Survey on prediction models of applications for resources provisioning in cloud. *J. Netw. Comput. Appl.* **82**, 93–113 (2017). <https://doi.org/10.1016/j.jnca.2017.01.016>
3. Liu, C., Liu, C., Shang, Y., Chen, S., Cheng, B., Chen, J.: An adaptive prediction approach based on workload pattern discrimination in the cloud. *J. Netw. Comput. Appl.* **80**, 35–44 (2017). <https://doi.org/10.1016/j.jnca.2016.12.017>. ISSN 1084–8045
4. Calzarossa, M.C., Massari, L., Tessera, D.: Workload characterization: a survey revisited. *ACM Comput. Surv.* **48**(3), 1–43 (2016). <https://doi.org/10.1145/2856127>. Article 48
5. Kousouris, G., Cucinotta, T., Varvarigou, T.: The effects of scheduling, workload type and consolidation scenarios on virtual machine performance and their prediction through optimized artificial neural networks. *J. Syst. Softw.* **84**(8), 1270–1291 (2011). <https://doi.org/10.1016/j.jss.2011.04.013>. ISSN 0164-1212
6. Sadeka, I., Jacky, K., Kevin, L., Anna, L.: Empirical prediction models for adaptive resource provisioning in the cloud. *Future Gener. Comput. Syst.* **28**(1), 155–162 (2012). <https://doi.org/10.1016/j.future.2011.05.027>. ISSN 0167-739X
7. Litke, A., Tserpes, K., Varvarigou, T.: Computational workload prediction for grid oriented industrial applications: the case of 3D-image rendering. In: CCGrid 2005. IEEE International Symposium on Cluster Computing and the Grid, Cardiff, Wales, UK, vol. 2, pp. 962–969 (2005). <https://doi.org/10.1109/CCGRID.2005.1558665>
8. Calheiros, R.N., Masoumi, E., Ranjan, R., Buyya, R.: Workload prediction using Arima model and its impact on cloud applications' QoS. *IEEE Trans. Cloud Comput.* **3**(4), 449–458 (2015). <https://doi.org/10.1109/TCC.2014.2350475>
9. Zhang, Q., Yang, L.T., Yan, Z., Chen, Z., Li, P.: An efficient deep learning model to predict cloud workload for industry informatics. *IEEE Trans. Industr. Inf.* **14**(7), 3170–3178 (2018). <https://doi.org/10.1109/TII.2018.2808910>
10. Tan, J., Dube, P., Meng, X., Zhang, L.: Exploiting resource usage patterns for better utilization prediction. In: 2011 31st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, pp. 14–19 (2011). <https://doi.org/10.1109/ICDCSW.2011.53>
11. Cao, J., Zhang, Q., Shi, W.: Challenges and opportunities in edge computing. In: Edge Computing: A Primer. SCS, pp. 59–70. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-02083-5_5
12. Zhang, K., Zhu, Y., Leng, S., He, Y., Maharjan, S., Zhang, Y.: Deep learning empowered task offloading for mobile edge computing in urban informatics. *IEEE Internet Things J.* **6**(5), 7635–7647 (2019). <https://doi.org/10.1109/JIOT.2019.2903191>

13. Zeng, D., Gu, L., Pan, S., Cai, J., Guo, S.: Resource management at the network edge: a deep reinforcement learning approach. *IEEE Netw.* **33**(3), 26–33 (2019). <https://doi.org/10.1109/MNET.2019.1800386>
14. Yuan, Q., Li, J., Zhou, H., Lin, T., Luo, G., Shen, X.: A joint service migration and mobility optimization approach for vehicular edge computing. *IEEE Trans. Veh. Technol.* **69**(8), 9041–9052 (2020). <https://doi.org/10.1109/TVT.2020.2999617>
15. Borkowski, M., Schulte, S., Hochreiner, C.: Predicting cloud resource utilization. In: Proceedings of the 9th International Conference on Utility and Cloud Computing, New York, NY, USA, pp. 37–42, December 2016. <https://doi.org/10.1145/2996890.2996907>
16. Thonglek, K., Ichikawa, K., Takahashi, K., Iida, H., Nakasan, C.: Improving resource utilization in data centers using an LSTM-based prediction model. In: 2019 IEEE International Conference on Cluster Computing (CLUSTER), pp. 1–8, September 2019. <https://doi.org/10.1109/CLUSTER.2019.8891022>
17. Baig, S.-R., Iqbal, W., Berral, J.L., Carrera, D.: Adaptive sliding windows for improved estimation of data center resource utilization. *Futur. Gener. Comput. Syst.* **104**, 212–224 (2020). <https://doi.org/10.1016/j.future.2019.10.026>
18. Baig, S., Iqbal, W., Berral, J.L., Erradi, A., Carrera, D.: Adaptive prediction models for data center resources utilization estimation. *IEEE Trans. Netw. Serv. Manage.* **16**(4), 1681–1693 (2019). <https://doi.org/10.1109/TNSM.2019.2932840>
19. Kaur, G., Bala, A., Chana, I.: An intelligent regressive ensemble approach for predicting resource usage in cloud computing. *J. Parallel Distrib. Comput.* **123**, 1–12 (2019). <https://doi.org/10.1016/j.jpdc.2018.08.008>
20. Yaqub, M., et al.: State-of-the-art CNN optimizer for brain tumor segmentation in magnetic resonance images. *Brain Sci.* **10**(7) (2020). <https://doi.org/10.3390/brainsci10070427>
21. GitHub, T. Pagoulatou. <https://github.com/titapag/HBPSHPO.git>
22. He, Y., Ma, W.J., Zhang, J.P.: The parameters selection of PSO algorithm influencing on performance of fault diagnosis. In: MATEC Web Conference, vol. 63, p. 02019 (2016). <https://doi.org/10.1051/matecconf/20166302019>
23. Brochu, E., Cora, V.M., de Freitas, N.: A Tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. [arXiv:1012.2599](https://arxiv.org/abs/1012.2599) [cs], December 2010
24. GitHub, G. Rodola', giampaolo/psutil. <https://github.com/giampaolo/psutil>
25. GitHub, A. K. Mortensen, anderskm/gputil. <https://github.com/anderskm/gputil>
26. Violos, J., Psomakelis, E., Tserpes, K., Aisopos, F., Varvarigou, T.: Leveraging user mobility and mobile app services behavior for optimal edge resource utilization. In: Proceedings of the International Conference on Omni-Layer Intelligent Systems, Crete, Greece, pp. 7–12, May 2019. <https://doi.org/10.1145/3312614.3312620>
27. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, August 2016. <https://doi.org/10.1145/2939672.2939785>
28. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Cortes, C., Lawrence, N.D., Lee, D.D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems, vol. 28, pp. 2962–2970. Curran Associates Inc. (2015)
29. Violos, J., Psomakelis, E., Danopoulos, D., Tsanakas, S., Varvarigou, T.: Using LSTM neural networks as resource utilization predictors: the case of training deep learning models on the edge. Presented at the 17th International Conference on the Economics of Grids, Clouds, Systems and Services (GECON), Zenodo. <https://doi.org/10.5281/zenodo.4274441>

30. Schmitt, M., Wanka, R.: Particle swarm optimization almost surely finds local optima. *Theoret. Comput. Sci.* **561**, 57–72 (2015). <https://doi.org/10.1016/j.tcs.2014.05.017>
31. Berkenkamp, F., Schoellig, A.P., Krause, A.: No-regret Bayesian optimization with unknown hyperparameters. *J. Mach. Learn. Res.* **20**(50), 1–24 (2019)



Approaching Deep Convolutional Neural Network for Biometric Recognition Based on Fingerprint Database

Md. Saiful Islam^(✉), Tanhim Islam, and Mahady Hasan

Fab Lab, Independent University Bangladesh, Dhaka, Bangladesh
mahady@iub.edu.bd

Abstract. Fingerprint dataset is one of the most broadly implemented and broadcasted biometrics for the derivation of individual feature identification. Fingerprint dataset performs in multiple approaches, such as applying query by image content techniques, reviewing criminal offenders, surveillance, taking a difficult decision, searching immediately, and anthropological research because of the uniqueness and persistence of the fingerprint dataset. Here in this research signifies an efficient way of identifying two key biological features: blood group and gender distinguish, based on the fingerprint dataset, applying Deep Convolutional Neural Networks (D-CNNs). The proposed model contains a modified approach of D-CNN and is trained and developed on a self-built fingerprint dataset. Thus, the algorithm applied here aims to observe how prominent the model performs for the custom-built dataset. The proposed model of D-CNN approach proved to be an improved technique and reaches an accuracy of around 99.968% based on the fingerprint images by the individuals for the identification of blood group and gender.

Keywords: Deep convolutional neural network · Blood group · Gender identification · Fingerprint · Machine learning

1 Introduction

Fingerprints are the unique biological feature of living beings which consist of loops, whorls and arches patterns of ridges. In each loop pattern, ridges enter and exit on the same side. Ridges look like circular in the whorls pattern, and in the arches pattern, ridges enter and exit separately [1]. As a combination of these parameters, the individual fingerprint results in a unique biological data. Fingerprints are totally uniform that no more than one individual have the same identical fingerprint. Survey study results propose that there is a one in 64 billion possibilities that two fingerprints will match with each other [2]. Focusing on this distinctive attribute, Fingerprints are ideal for the purpose of identifying other biological characteristics such as gender and blood group of individuals.

Thus, the proposed D-CNN model focuses on deriving gender and blood group from fingerprint dataset. As fingerprint data is rational to collect and analyze because of its

versatility of the data type, we aim to utilize this property. The blood group determination is done by the noninvasive method by considering two procedures, understanding the correlation of blood groups and fingerprints of the participants and considering the fingerprints of the participants only [3]. Individual identification is required in many different areas such as security and video monitoring employment, emergency medical services, biometric data analysis etc. An individual can be recognized by various biometric features, such as physical appearance, vocal sound, stature, and shape. Among the biological features, gender is one of the most fundamental properties, that distinguish people. Also, the blood group is a vital key property of individual personnel. Fingerprinting can be considered the best parameter for distinguishing between individuals and for tracking people independently. Specially, in the sector of criminal investigations, the fingerprint characteristics can lessen the male-female determination challenges and restrict the number of times it takes to recognize a suspect among a greater group of people. The features of fingerprints can be applied in distinguishing among individuals by their gender and blood group; therefore, it fastens the decision processes of unknown suspects. Moreover, this can escort forensic investigators to the exact identification of a defendant when matching the suspect's fingerprints among a wide number of potential matches in the fingerprint databases [2].

In this paper, we proposed a model approaching D-CNNs algorithm for identifying blood group and gender based on our custom-built fingerprint dataset. The algorithm used in this model contains modification on working principles and layer dimensions. For the dataset, a group of individuals biometric fingerprint data has been collected with appropriate data usage authority and developed into a well-functional data structure for adaptability with the proposed algorithm model. The established model has displayed a significant result and in this paper the result is also compared to the previous approaches and demonstrated as a better application on this sector.

2 Literature Review

Fingerprint recognition is demonstrated in a lot of research studies. However, an inadequate number of studies is published for gender and Blood group identification. Gornale, S. et al. [4], explained a method to determine the gender of an individual based on the fingerprint database. To establish their study, they worked with two distinct techniques like discrete wavelet transform and Gabor filter for grouping gender and considered less than a hundred participants for their dataset. In the end, they received 85% and 87% accuracy rate in terms of classification and received 97% rate with the implementation of the Gabor filter.

Lian, H. C. et al. [5], applied Local Binary Pattern and Support Vector Machine based on the CAS-PEAL face dataset for the male-female recognition. In Support Vector Machine implementation, they have considered the polynomial kernel. Finally, their study received 96.75% accuracy there.

Tom, R. J. et al. [6], used the blending of Discrete Wavelet Transform and Principal Component Analysis based on 400 fingerprints (where the same quantity of male-female dataset is considered) for gender identification. They have observed that the rise of the performance of a model is proportional to the extension of the quantity of a dataset.

Sun, Z. et al. [8], represented a hybrid approach with the combination of Principal Component Analysis (PCA), Generic Algorithm (GA) and Neural Network (NN) where they proposed an automatic feature subset selection method. For obtaining essential characteristics of an image vector applied PCA and to determine each subset of that character, used GA. In the end, they increase the accuracy from 82.3% to 88.7% by using NN.

Verma, M. et al. [9] applied Support Vector Machine for the classifier of definite characteristics of ridges. They proposed a correlation linking fingerprints and gender applying the definite characteristics of ridges alike density, width, and thickness to valley thickness and secured above 91% accuracy rate.

Rattani, A. et al. [10], observed Local Binary Pattern, Local Phase Quantization, Binarized Statistical Image Features, and Local Ternary Pattern based on fingerprint database for identification of male-female. While examining different techniques they have found that in terms of the blurred image, Local Phase Quantization performs better and in terms of incomplete image, Binarized Statistical Image Features performs better. They also recommended that, in this section researchers should focus more.

Kaur, R. et al. [11], explored the combination of Fast Fourier Transform, Discrete Cosine Transform, and Power Spectral Density to determine gender. They also worked with the fingerprint dataset for attaining their research purpose. Here they have considered around 220 datasets to predict gender from their proposed model.

3 Material and Method

3.1 Convolutional Neural Network

Convolutional Neural Network (CNN), refers to a deep learning algorithm which takes images as input, assigns weights or some measuring values to various properties of the image differing among individual images. The motivation for choosing this algorithm is the requirement of pre-processing in a ConvNet is low cost than other classification algorithms.

Previously released methodologies contain filters which are self-manipulated with sufficient training, while ConvNet can learn automatically. There are associations between ConvNet and Neurons of the human brain. They are similar to their connectivity construction. Furthermore, ConvNet is influenced by the adjustment of a certain part of the human brain that interprets visual knowledge.

- Convolutional Layer: Convolutional layer is the principal base toward assembling a convolutional neural network. A sequence of filters (weights) forms every single convolutional layer. Every individual convolutional layer performs scalar product with the prior convolutional layer [12].
- Max-Pooling Layer: the max-pooling is another base towards assembling a convolutional neural network. While receiving the outcome of a prior convolutional layer, max-pooling subsets the samples with the consideration of the highest value [12].
- Fully Connected Layer: The contribution of the fully connected layer is, it combines a collection of neurons with their previous layer's neurons [12].

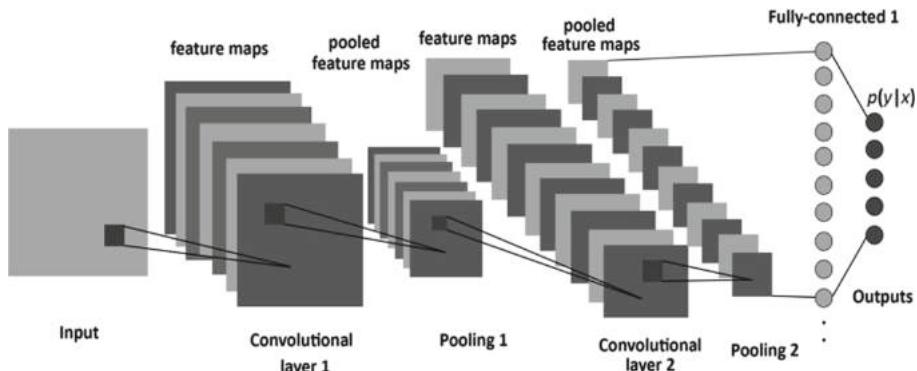


Fig. 1. Convolutional neural network [12]

3.2 Proposed Convolutional Neural Network

We have considered the fingerprint image of size 340×340 as an input of this CNN model. For our modification on the D-CNN approach, we have customized the dimensions of different layers of the convolutional neural network. In the first layer, the convolutional layer where 32 weights of dimension $(3, 3)$ are applied with a rectified linear unit activation function. The max-pool layer has a pool size of 2×2 is which is implemented to the image. Besides, the second convolutional layer where 64 weights of dimension $(3, 3)$ are implemented including a rectified linear unit activation function. The second max pool layer has a pool size of 2×2 is implemented to the image. Then, the third convolutional layer where 128 weights of dimension $(3, 3)$ are implemented including a rectified linear unit activation function. The third max-pool layer has a pool size of 2×2 is implemented to the image. Moreover, the fourth convolutional layer where 128 weights of dimension $(3, 3)$ are implemented including a rectified linear unit activation function. Then, the first up sampling layer with $(2, 2)$ dimension is implemented in the image for the fourth convolutional layer. Furthermore, the fifth convolutional layer where 128 weights of dimension $(3, 3)$ are implemented with a rectified linear unit activation function. Following, the second up sampling layer with $(2, 2)$ dimension is implemented in the image for the fifth convolutional layer. Finally, concerning the result convolutional layer where 1 filters of dimension $(3, 3)$ are implemented including a sigmoid activation function. The proposed CNN is displayed in Table 1 and illustrated in Fig. 1.

4 Dataset and Result Analysis

As there is a dearth of approved fingerprint datasets for gender and blood group identification, we have come up with our own dataset. To build a dataset, we have collected 1000 fingerprints of people comprising different ages and gender and then scanned from their ink prints. An example is displayed in Fig. 2. In this dataset, we have considered parameters of gender and blood group. The dataset was created in the laboratory with ink prints and following scanned with an L120 scanner device. We have selected the high-quality image option with accurate image labeling for each image of the fingerprint.

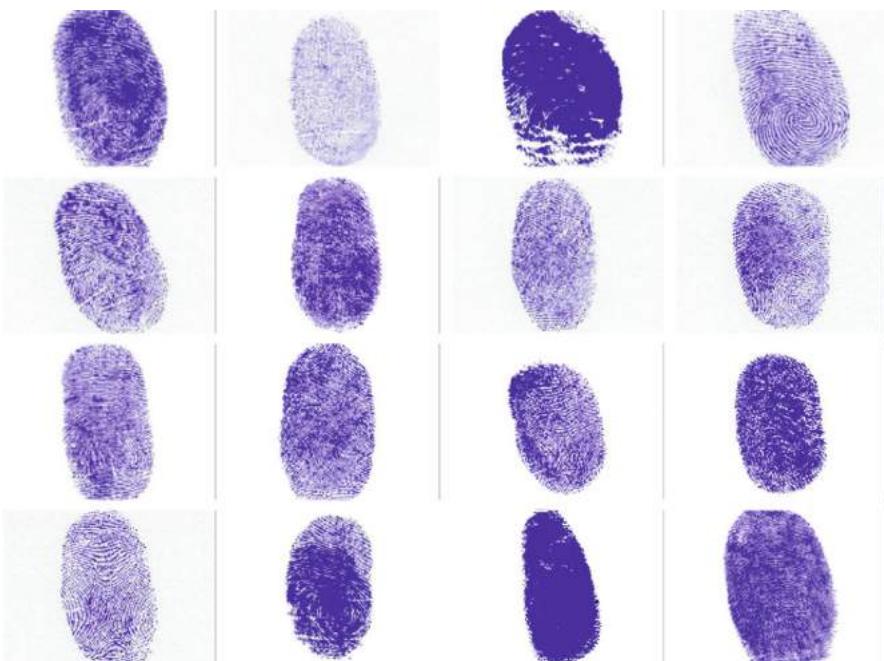


Fig. 2. Dataset sample

In total, we have collected 5000 fingerprints from the user (1000 users \times 5 images per user). Later, 1000 fingerprints have considered as 1000 samples in this research. Here, from 1000 samples, 80% dataset is considered for the training, and 20% dataset is considered for the testing. The input image data are reshaped and converted into data arrays that refer to the data preprocessing for the implementation. In Fig. 3, the data array is presented in grayscale display mode.

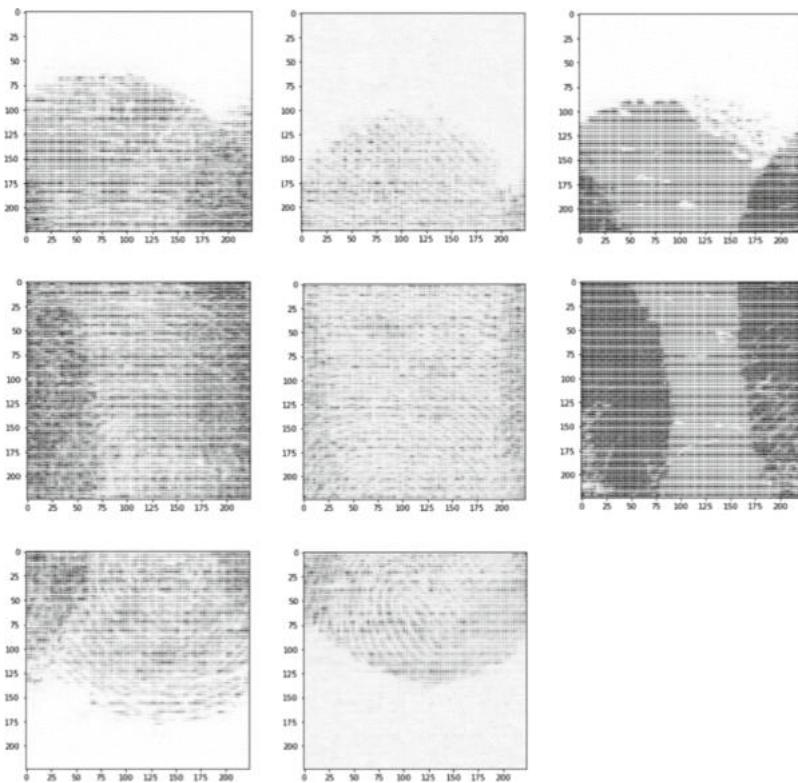


Fig. 3. Fingerprint dataset at image processing step (grayscale display)

In our granted dataset, out of 1000 fingerprints, 507 samples are from male candidates, and 493 samples are from female candidates. Figure 4 shows the number of different blood groups for the male and female individual which we have considered in our dataset. It is clearly seen that for blood group O+, 335 samples have been collected in total, where 195 samples are from the male person, and 140 samples are from the female person. Again, for blood group A+, 244 samples have collected in total, where 124 samples are from the male person, and 120 samples are from the female person. Besides, for blood group B+, 239 samples have collected in total, where 139 samples are from the male person, and 100 samples are from the female person. Finally, for blood group AB+, 182 samples have collected in total, where 120 samples are from the male person, and 62 samples are from the female person.

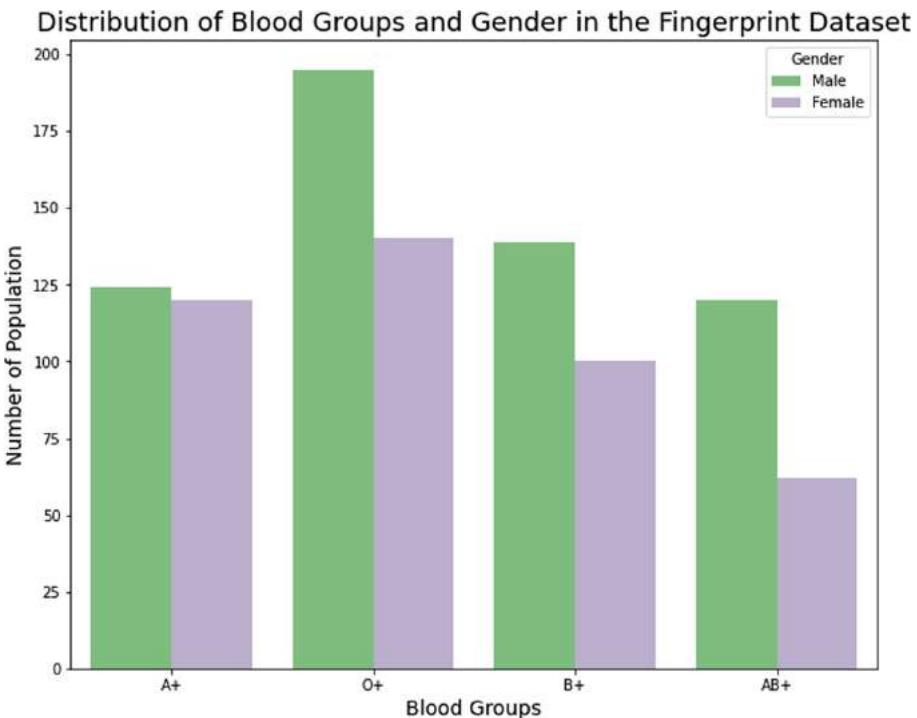


Fig. 4. Distribution of blood groups (O+, A+, B+ and AB+) and gender in the fingerprint dataset.

We have accomplished our research according to the proposed convolutional network model, which is mentioned in Sect. 3 and Table 1. A deep convolutional neural network applied in this result.

Table 1. Proposed convolutional neural network architecture

Layer (type)	Output Shape	Param #
conv2d_25 (Conv2D)	(None, 340, 340, 32)	320
max_pooling2d_9 (MaxPooling2)	(None, 170, 170, 32)	0
conv2d_26 (Conv2D)	(None, 170, 170, 64)	18496
max_pooling2d_10 (MaxPooling)	(None, 85, 85, 64)	0

(continued)

Table 1. (*continued*)

Layer (type)	Output Shape	Param #
conv2d_27 (Conv2D)	(None, 85, 85, 128)	73856
conv2d_28 (Conv2D)	(None, 85, 85, 128)	147584
up_sampling2d_9 (UpSampling2)	(None, 170, 170, 128)	0
conv2d_29 (Conv2D)	(None, 170, 170, 64)	73792
up_sampling2d_10 (UpSampling)	(None, 340, 340, 64)	0
conv2d_30 (Conv2D)	(None, 340, 340, 1)	577

Total params: 314,625

Trainable params: 314,625

Non-trainable params: 0

Here in Table 2, batch size 128 and 200 of epochs have been implemented. Initially, training loss was nearby 0.2207, and training accuracy was approximately 99.7793%. After completion of 200 epochs, training loss has reduced (stayed at 0.0216) and training accuracy was around 99.9784%.

Table 2. Training accuracy and training loss of the proposed method

Epoch	Training accuracy (%)	Training loss
1	99.7793	0.2207
2	99.761	0.2390
3	99.8047	0.1953
4	99.8369	0.1631
5	99.8455	0.1545
196	99.9573	0.0427
197	99.9574	0.0426
198	99.9575	0.0425
199	99.9578	0.0422
200	99.9784	0.0216

In this research paper, our proposed deep convolutional neural network model has achieved 99.968% test accuracy with an amount Loss of 0.03139 for blood group and gender identification based on 1000 fingerprint images which are displayed in Table 3.

Table 3. Test accuracy of the proposed method

Model	Testing accuracy	Testing loss
D-CNN proposed model	99.96861	0.03139

Several fingerprint recognition methods are used to identify gender and blood group. While the traditional methods cannot provide satisfactory results in the case of unavailability of some features of fingerprint, our model is capable of predicting and identifying one's fingerprint even when some features are not found. Besides this, our test accuracy is more significant compared to the other previously used techniques as shown in Table 4. Because of the testing error is close to the training error, we can conclude that our model is a robust one and prone to overfitting.

Table 4. Test accuracy for gender classification

Method	Accuracy
RTVTR, SVM	91%
CNN	85.7%
SVM with polynomial kernel	94.08%
Our model	99.97%

5 Conclusion

Our research proposes a D-CNNs approach based on fingerprint recognition architecture to show the best way of identifying blood group and gender based on fingerprints by the individual. Moreover, this algorithm performs for a custom-built dataset. The proposed D-CNNs model performs a test accuracy of around 99.97% for the given dataset. One of the limitations of this project is the small size of training data: the model test set accuracy would have been greater with a larger dataset. In the future, we will use a larger dataset for better accuracy and reliability. Our future work will be concerned about integrating spatial domain, singular value decomposition analysis and other approaches to find different parameters like ethnicity, height, skin color, and age. Additionally, different features will be included to assist in blood group, and gender classifications which will perform better with other machine learning approaches. The scope of our project is wide in biometrics and forensic anthropology.

References

1. How Stuff Work (2018). How Fingerprinting Works. <https://science.howstuffworks.com/fingerprinting1.htm>. Accessed 21 Mar 2019

2. Dantcheva, A., Elia, P., Ross, A.: What else does your biometric data reveal? A survey on soft biometrics. *IEEE Trans. Inf. Forensics Secur.* **11**(3), 441–467 (2015)
3. Chaudhary, S., Deuja, S., Alam, M., Karmacharya, P., Mondal, M.: Fingerprints as an alternative method to determine ABO and Rh blood groups. *JNMA J. Nepal Med. Assoc.* **56**(208), 426–431 (2017)
4. Gornale, S., Patil, A., Veersheety, C.: Fingerprint based gender identification using discrete wavelet transform and gabor filters. *Int. J. Comput. Appl.* **975**, 8887 (2016)
5. Lian, H.-C., Lu, B.-L.: Multi-view gender classification using local binary patterns and support vector machines. In: Wang, J., Yi, Z., Zurada, J.M., Lu, B.-L., Yin, H. (eds.) *ISNN 2006. LNCS*, vol. 3972, pp. 202–209. Springer, Heidelberg (2006). https://doi.org/10.1007/117600_23_30
6. Tom, R.J., Arulkumaran, T., Scholar, M.E.: Fingerprint based gender classification using 2D discrete wavelet transforms and principal component analysis. *Int. J. Eng. Trends Technol.* **4**(2), 199–203 (2013)
7. Sciences Truck (2018). Fingerprint Patterns: Identifying the Different Types Easily. <https://sciencestruck.com/identifying-types-of-fingerprints-patterns>. 18 Apr 2019
8. Sun, Z., Yuan, X., Bebis, G., Louis, S.J.: Neural-network-based gender classification using genetic search for eigen-feature selection. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN 2002* (Cat. No. 02CH37290), vol. 3, pp. 2433–2438. IEEE, May 2002
9. Verma, M., Agarwal, S.: Fingerprint based male-female classification. In: Corchado, E., Zunino, R., Gastaldo, P., Herrero, Á. (eds.) *Proceedings of the International Workshop on Computational Intelligence in Security for Information Systems CISIS'08*, pp. 251–257. Springer Berlin Heidelberg, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88181-0_32
10. Rattani, A., Chen, C., Ross, A.: Evaluation of texture descriptors for automated gender estimation from fingerprints. In: Agapito, L., Bronstein, M.M., Rother, C. (eds.) *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*, pp. 764–777. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-16181-5_58
11. Kaur, R., Mazumdar, S.G.: Fingerprint based gender identification using frequency domain analysis. *Int. J. Adv. Eng. Technol.* **3**(1), 295 (2012)
12. Tivive, F.H.C., Bouzerdoum, A.: A gender recognition system using shunting inhibitory convolutional neural networks. In: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 5336–5341. IEEE, July 2006
13. O’Shea, K., Nash, R.: An Introduction to Convolutional Neural Networks. ArXiv e-prints (2015)



Optimizing the Neural Architecture of Reinforcement Learning Agents

N. Mazyavkina, S. Moustafa, I. Trofimov^(✉), and E. Burnaev

Skolkovo Institute of Science and Technology, Moscow, Russia
{n.mazyavkina,samir.mohamed,ilya.trofimov,e.burnaev}@skoltech.ru

Abstract. Reinforcement learning (RL) enjoyed significant progress over the last years. One of the most important steps forward was the wide application of neural networks. However, architectures of these neural networks are quite simple and typically are constructed manually. In this work, we study recently proposed *neural architecture search (NAS)* methods for optimizing the architecture of RL agents. We create two search spaces for the neural architectures and test two NAS methods: Efficient Neural Architecture Search (ENAS) and Single-Path One-Shot (SPOS). Next, we carry out experiments on the Atari benchmark and conclude that modern NAS methods find architectures of RL agents outperforming a manually selected one.

Keywords: AutoML · Neural architecture search · Reinforcement learning · Atari

1 Introduction

Over the last several years, deep learning (DL) has experienced enormous growth in popularity among the researchers from both the academia and the industry. Moreover, each of the separate tasks solved by the DL methods requires its own approach, one of the most important aspects of which is the choice of the neural network's (NN) architecture. In this case, it is essential to demonstrate good expertise and experience in the problem's field. However, even then, the chosen architecture may not give any acceptable results until various heuristics and tricks will be applied to its construction. This motivated the emergence of the *neural architecture search (NAS)* field, which focuses on automating the ways to find the optimal architecture for the specific tasks.

Another family of methods that has been gaining popularity is reinforcement learning (RL) and deep reinforcement learning (deep RL), in particular. It consolidates a vast collection of machine learning methods, designed to solve a variety of Markov-Decision-Process-like problems. Over the last several years the successes of RL and deep RL has been frequently demonstrated by the research community: from better-than-human performance in ATARI [17], DOTA 2 [15],

N. Mazyavkina and S. Moustafa—Equal contribution.

Go [23] to robotic manipulation [8]. In deep RL, neural networks are usually used to approximate a *value* function, in the case of the value-based methods, or a *policy* function, in the case of policy gradient methods. Moreover, actor-critic RL algorithms [16, 24] combine these two NN approximations, in order to gain even better performance. Consequently, finding a suitable NN architecture is also a vital part of designing RL experiments.

In this work, we are going to explore deep RL as a new application of NAS, i.e. deriving well-performing NN architectures for RL tasks. The motivation for the research in this direction is the following:

1. Only a single NN architecture is often chosen for many common benchmarks such as ATARI [17] and MuJoCo [25], despite of them consisting of a big number of different environments. Hence, automatically finding a suitable network for each of the environments may lead to better results;
2. NAS can be useful in the cases of more complicated environments with bigger state and action spaces, where a more complicated deeper network might be required.

Early NAS methods [31, 32] required training of numerous neural architectures. However, even training of one RL agent takes a significant amount of time. In this paper, we limit ourselves to fast *one-shot* methods, which perform architecture search in the time not significantly larger than the training time of a single neural network. Most of the NAS methods were developed for computer vision applications, the major part – for the object classification problem. At the same time, reinforcement learning is quite a different problem. The performance of an RL agent, that is, average reward, is not differentiable like the cross-entropy of object classification. Thus, only few popular NAS methods are suited for RL. In this work, we evaluate ENAS [32] and SPOS [9].

The contribution of our paper is the following: we experimentally prove that modern one-shot NAS methods can be successfully applied for optimizing the neural architecture of RL agents. The source code is publicly available from https://github.com/NinaMaz/NAS_RL_torch.

2 Related Work

Early neural architecture search (NAS) approaches treated this problem as a black-box optimization, that is, search over a discrete domain of architectures. Such methods are quite general but require training of numerous architectures and vast computational resources. One of the first proposed methods of this kind [31, 32] used reinforcement learning for the optimization process itself. Architecture creation, layer by layer, was done by an RL agent. Thus, the reward was the performance of the constructed network. Other works proposed evolutionary optimization [20], Bayesian optimization based on Gaussian processes [10], bayesian performance predictors based on architecture features [22, 27]. Black-box optimization enjoy speedup from multi-fidelity methods [26]. Several benchmarks for NAS were developed [6, 11, 30].

The later family of methods – *one-shot NAS* – gone beyond black-box optimization and utilized the structure a neural network. These methods involve the *supernet*, which contain all the architectures from the search space as its subnetworks. Thus, all the architectures share weights of some of the blocks. The architecture search in the supernet is performed simultaneously with the training of networks themselves. The one-shot methods are: ENAS [19], numerous modifications of DARTS [3–5, 13, 14, 29], single path one-shot [9], random search with weight sharing [1, 12].

Most of the existing research focuses on problems from computer vision and linguistics. There are no papers about applications of modern NAS methods to RL to the best of our knowledge.

3 Reinforcement Learning Methods

In our experiments, we have used reinforcement learning for training both ENAS controller and sampled child networks. On the other hand, SPOS does not use a trainable controller for architecture sampling and, hence, the RL methods, mentioned in this section, do not concern it.

An LSTM controller, used in the ENAS framework, is trained with REINFORCE [28] algorithm. REINFORCE belongs to a group of policy-based methods, which focuses on the straightforward approximation of the optimal policy, via calculating the direct gradient of the parameterized objective function J :

$$\leq_{\theta} J_{\theta} = \mathbb{E}_t \left(\leq_{\theta} \log \pi_{\theta}(a_t | s_t) R_t \right).$$

In our case, θ are the parameters of the neural network, outputting the logits, from which the actions a_t can be derived; π is the policy, s_t are the states, R_t is the sum of the discounted rewards collected so far. Specifically, in the case of REINFORCE algorithm, the update to the parameters θ takes the form:

$$\theta * \theta + \gamma^t R_t \leq_{\theta} \log \pi_{\theta}(a_t | s_t),$$

where γ is the discount factor. In order to reduce the variance of the gradient estimation from the formula above, we subtract the moving average baseline from the discounted reward function.

In terms of the training process of the child networks, we use another policy-based method - Proximal Policy Optimization (PPO) [21] to update their parameters. The objective function for PPO has the following form:

$$J_{\theta}^{PPO} = \mathbb{E}_t \left(\min \left[ratio_t(\theta) A_t, clip(ratio_t(\theta), 1 + \epsilon, 1 - \epsilon) A_t \right] \right),$$

where A_t is the advantage function, $ratio$ is the probability ratio under the new and old policies, ϵ is the clipping parameter.

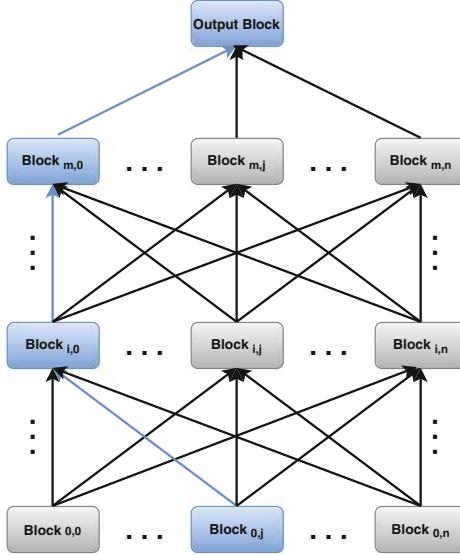


Fig. 1. Supernet architecture for a generic design. Each block can be a part of neural network, and each connection between sequence of blocks can be a complete architecture. The blue path represent a complete architecture.

4 Neural Architecture Search Methods

4.1 Adaptation to Reinforcement Learning

Most of the existing research on NAS is dedicated to computer vision (particularly, object classification) or computational linguistics applications. We adapt the existing one-shot NAS methods to reinforcement learning. One-shot methods assume the *supernet* which contain all the architectures from the search space as its subgraphs (Fig. 1). All the architectures share weights of some of the blocks. That is, each layer in the supernet i has a list of $Block_i$ options, only one option can be selected for a particular subnetwork. Such simplification was proposed to reduce the co-adaptation between blocks. The whole search space is $\mathcal{A} = Block_0 \times \dots \times Block_n$. Some initial or final layers of the supernet can be fixed and not contain choice blocks.

One-shot methods typically contain a *fitting* stage. During the *fitting* stage, the subnetwork $\alpha \in \mathcal{A}$ is sampled by some rule and its weights $\Theta(\alpha)$ are updated by a SGD-like step for a batch of data B

$$\Theta(\alpha) * \Theta(\alpha) - \eta \leq_{\Theta(\alpha)} \sum_{i \in B} \ell(y_i, N(\alpha, \Theta(\alpha), x_i)), \quad (1)$$

where $\ell(\cdot)$ is the loss function, $N(\alpha, \theta, x_i)$ is the network of the architecture α having weights $\Theta(\alpha)$. The *evaluation* of the architecture α typically involves calculation of performance (accuracy) on the validation dataset D_{val}

$$\frac{1}{|D_{val}|} \sum_{i \in D_{val}} [y_i = N(\alpha, \Theta(\alpha), x_i)]. \quad (2)$$

We adapt one-shot methods to RL in the following way. In our experiments, the neural network $N(\alpha, \Theta(\alpha), x_i)$ corresponds to a policy $\pi_{\Theta(\alpha)}$. Instead of SGD-like step (1) we do the step of PPO

$$\Theta(\alpha) * \Theta(\alpha) - \eta \leq_{\Theta(\alpha)} J_{\Theta(\alpha)}^{PPO}. \quad (3)$$

The performance of the network is estimated by $\mathbb{E}_t[R_t[\pi_{\Theta(\alpha)}]]$ instead of (2).

4.2 Efficient Neural Architecture Search (ENAS)

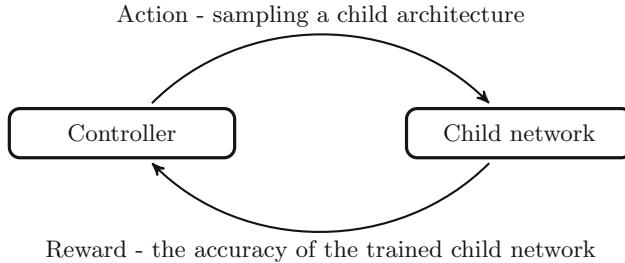


Fig. 2. A general framework of neural architecture search (NAS)

Efficient Neural Architecture Search (ENAS) [19] is a NAS method, used in our experiments to find a well-performing neural network in the ATARI games environment. ENAS consists of a controller, which samples child models, a search strategy and an evaluation strategy.

In the case of ENAS, the controller is an LSTM network that outputs a one-hot-encoded architecture of a child model. The controller's inputs are the previous step's architecture and the reward it has received. The possible choices for a set of child architectures are determined by the search space. The particular variants of the search spaces that we have used are covered in Sect. 5.2.

The authors of [19, 31] have proposed to train an ENAS controller by using RL. In our experiments, we have employed a REINFORCE algorithm, which has also been used in the original paper to update the controller's network parameters.

In the original paper, due to the different nature of the problems ENAS has been tested on, the child networks, sampled by the controller, are trained until convergence. However, this becomes difficult when the RL environments are considered – the agents usually require longer training times, and the stochasticity of the environments makes the training process much more unpredictable. In our work, we will demonstrate that despite the aforementioned problems, the

number of child training timesteps, chosen for our experiments, is still sufficient to determine which networks are better-performing than others.

Finally, the overall sequence of “sampling a child architecture - training a child model - feeding the resulting reward to LSTM controller” (see Fig. 2) defines a single epoch of ENAS training. In the previous works on iterative RL NAS methods [31, 32], the fact that such an epoch will take a long time to compute, has limited these methods’ practicality in regard to the real-life problems. The authors of ENAS, however, have come up with the solution to this problem by sharing the weights of all of the child models. This way, the ENAS becomes much more efficient than its predecessors, and, therefore, much more suitable for RL problems.

4.3 Single-Path One-Shot with Uniform Sampling (SPOS)

The method “Single-path one-shot with uniform sampling” was proposed in [9]. The SPOS method assumes two steps: 1) supernetwork fitting 2) best architecture selection. The distinctive feature of SPOS is that subnetworks are sampled from the supernetwork uniformly at random.

Thus, weights Θ^* of the supernetwork are the solution of the following problem

$$\Theta^* = \operatorname{argmin}_{\Theta} \mathbb{E}_{\alpha \sim P}[L_{train}(N(\alpha, \Theta(\alpha)))],$$

where $L_{train}(\cdot)$ is the train loss, P - the uniform distribution. During the architecture selection phase, the best subnetwork α^* is selected by the validation accuracy Acc_{val}

$$\alpha^* = \operatorname{argmax}_{\alpha \in \mathcal{A}} Acc_{val}(N(\alpha, \Theta^*(\alpha))). \quad (4)$$

This step requires only inference for the validation data. In the original paper [9], an evolutionary optimization was used to solve (4) since the search space was huge. Instead of evolutionary optimization, we do the full search since our search spaces are small.

The adaptation of SPOS to RL is done as described in the Sect. 4.1. The subnetwork $N(\alpha, \Theta(\alpha))$ corresponds to a policy $\pi_{\Theta(\alpha)}$. Thus, SPOS for optimizing the neural architecture of the RL agent solves the following problem

$$\Theta^* = \operatorname{argmax}_{\Theta} \mathbb{E}_{\alpha \sim P} \mathbb{E}_t[R_t[\pi_{\Theta(\alpha)}]], \quad (5)$$

$$\alpha^* = \operatorname{argmax}_{\alpha \in \mathcal{A}} \mathbb{E}_t[R_t[\pi_{\Theta^*(\alpha)}]]. \quad (6)$$

5 Experiments

5.1 Atari Environment

In our experiments, we used the Open AI Gym framework [2], particularly – *Breakout* and *Freeway* Atari environments. We chose the *Breakout* because it’s

a popular benchmark, having moderate standard deviation of RL agent’s reward (401.2 ± 26.9 , [18]). In opposite to the *Breakout*, the *Freeway* environment has very low relative standard deviation of reward (30.3 ± 0.7 , [18]). The reward of RL agent with random behavior for *Breakout* and *Freeway* is nearly zero so we can make sure that our policy network makes non-stochastic behavior.

We have trained the child networks in the manner described in [7], i.e., we have used 8 agents, sharing a policy, trained simultaneously in 8 environments with PPO, in order to collect the trajectories for the policy update. Each of these agents is trained for 128 steps. After that, the controller collects the architecture’s rewards. The controller’s policy is updated every ten steps using the REINFORCE algorithm. Overall, the number of training steps for one experiment equals 10 million. More implementation details are in Appendix C.

It is important to note that the ‘scratch’ experimental results demonstrate lower reward values than the ones reported in the original PPO paper [21]. This is due to the fact that in our experiments we have used a smaller number of training timesteps than the classical version of PPO (10M vs. 40M). The reason for this has been that the main aim of our research focuses on investigating whether NAS has a positive effect on RL training process overall, and not on beating the existing ATARI baselines.

5.2 Search Spaces

The search spaces that we designed are the extension of the Nature-CNN architecture [18], where convolutional layers are followed by a linear layer which outputs the number of values equal to the size of our action space [18].

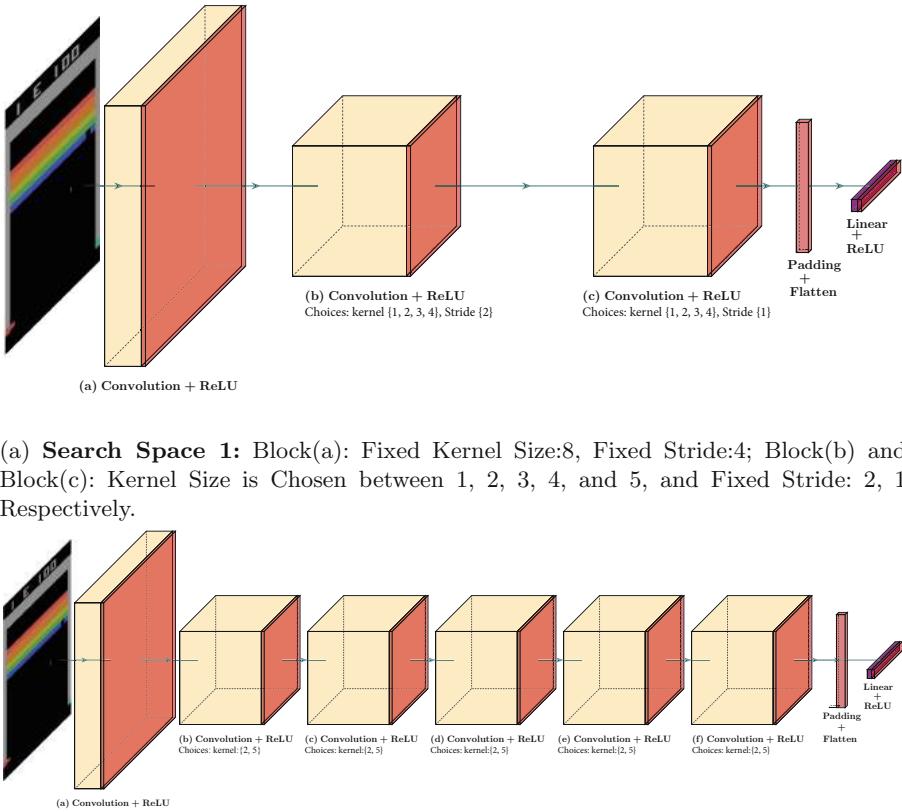
In order to facilitate the varying sizes of the layers’ parameters and, hence, enable the weight sharing, we use the following techniques in the overall design of the network:

1. **Convolution and max-pooling to the same size:** Map the input to one size every time after each convolutional and max-pooling layer.
2. **Padding to exact size:** Map the output of the last convolutional layer to the target size to be able to fix it during the flattening of the convolutions.

In our experiments we have covered two architectural search spaces:

Search Space 1: 25 Architectures. The architecture starts with a fixed convolutional layer with input channels equal to 4, and the output channels - to 32, kernel size - 8 and moving stride - 4. This layer is followed by two convolutional layers with output channels equal to 64 per each and with strides 1 and 2. The choices for kernel sizes are 1, 2, 3, 4, 5. The size of this search space is 5^2 .

Search Space 2: 32 Architectures. Instead of two convolutional layers (we do not take into account the first fixed convolutional layer, as we do not vary its kernel size) used in the search space 1, we have used 5 convolutional layers with output channels equal to 64, moving strides equal to 1. The choices for kernels are 2 and 5, which makes the size of this search space equal to 2^5 .



(a) **Search Space 1:** Block(a): Fixed Kernel Size:8, Fixed Stride:4; Block(b) and Block(c): Kernel Size is Chosen between 1, 2, 3, 4, and 5, and Fixed Stride: 2, 1 Respectively.

(b) **Search Space 2:** Block(a): Fixed Kernel Size:8, Fixed Stride:4; Block(b)- Block(f): Kernel Size is Chosen between 2 and 5, With Fixed Stride: 1 for All of them.

Fig. 3. Search spaces used in experiments.

The experimental architecture spaces that we have used can be seen in Fig. 3a and Fig. 3b, and also in Appendix A. The search spaces do not contain very deep architectures, having depth 7 at most.

5.3 Methodology

Firstly, we trained all the architectures from scratch (implementation details are in Appendix C). For all of those architectures we saved *mean reward* and *total reward* averaged over last 100 episodes. These metrics were used as a tabular benchmark, eliminating the need to train the same architecture multiple times. We used these metrics later to compare the performances of the found architectures by various NAS methods.

Both NAS methods under evaluation (ENAS, SPOS) share the same high-level structure:

1. Run neural architecture search method for the given search space \mathcal{A} ;
2. Select top-K architectures from the search space \mathcal{A} by a *proxy performance*;
3. Train these top-K architectures from scratch;
4. Return the best one by the *true performance*.

In our experiments, we selected top-3 architectures for the methods under evaluation. We repeated the search 4 times with different seeds and averaged results.

The *proxy performance* is a part of the NAS method, and it is fast to calculate. The calculation time of the proxy performance is negligible to the time of RL agent training from scratch. For ENAS, the proxy performance of an architecture is the probability of sampling this architecture by the controller. For SPOS, the proxy performance of an architecture is calculated from weights of this architecture in the supernet. Namely, the proxy performance is the mean reward of an agent with corresponding weights.

The *true performance* is the total/mean reward of an RL agent trained from scratch.

5.4 Random Search Baseline

As a simple baseline, we used the following random search algorithm:

1. Select K architectures from the search space \mathcal{A} at random;
2. Train these K architectures from scratch;
3. Return the best one by the *true performance*.

As for ENAS and SPOS, we selected top-3 architectures. The only difference with ENAS/SPOS methods is that architectures are selected at random instead of by *proxy performance*. We estimated the variance of the random search by repeating it for 1000 times.

5.5 Experiments with Larger Search Spaces

We have also tried to expand the search space by increasing the number of consecutive convolutional layers up to 9 and choosing a suitable kernel size and a number of output channels for each of them. However, the results were close to the ones received by random search, which can be caused by the fact that NAS can experience problems with increasingly complex search spaces. For that reason, we do not present the results of these experiments in the paper.

6 Discussion

Table 1 shows the results of our experiments. For the Breakout environment, ENAS performs better than random search on the search space 1, while SPOS performs better on the search space 2. For the Freeway environment, there is no clear benefit of NAS methods. Variance of the both of the methods are quite

Table 1. Reward mean and total reward of RL agents with various architectures. Each score is the mean of last 100 episodes.

		Search space 1		Search space 2	
		Breakout	Freeway	Breakout	Freeway
Random Search	Reward mean	54.7 ± 8.3	28.4 ± 1.0	33.1 ± 29.5	21.6 ± 0.2
	Total reward	147.2 ± 25.5	31.7 ± 0.8	105.7 ± 94.2	22.0 ± 0.2
Nature-CNN [18], reproduced	Reward mean	57.1	13.1	—	—
	Total reward	157.9	19.0	—	—
ENAS	Reward mean	61.4 ± 1.8	26.4 ± 1.1	30.7 ± 21.7	21.5 ± 0.1
	Total reward	161.1 ± 9.8	30.7 ± 1.3	91.4 ± 64.4	22.0 ± 0.2
SPOS	Reward mean	39.7 ± 18.6	29.6 ± 0.8	39.9 ± 41.0	21.7 ± 0.1
	Total reward	144.4 ± 55.0	29.4 ± 5.0	180.6 ± 72.5	22.0 ± 0.1

high. At the same time, SPOS is simpler since it does not contain an auxiliary controller network.

It is interesting to compare the performances of found architectures with the performance of manually selected Nature-CNN architecture [18] (see description in the Appendix B). The Nature-CNN architecture belongs to the search space 1. For the fair comparison, in Table 1, we report the rewards after training of the Nature-CNN architecture with our pipeline (Sect. 5.1). The architectures found by NAS methods outperform the manually selected Nature-CNN by a considerable margin.

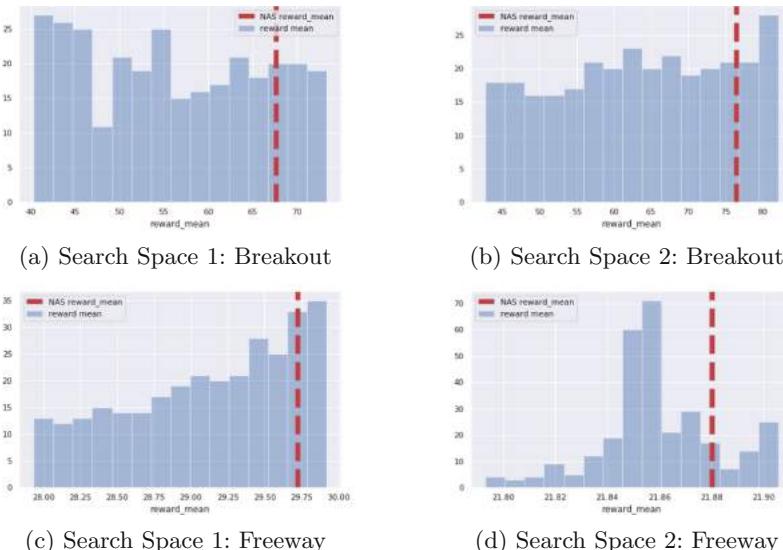


Fig. 4. Histograms for the reward mean of RL agents having different architectures from search spaces. The red vertical line depicts the best architecture found by NAS methods.

The existing research on RL methods typically uses the same architectures for different environments. However, we found that this is not optimal. Different architectures are optimal for different environments, see Appendix D.

Figure 4 shows histograms of RL agents’ mean rewards from different search spaces. The red vertical lines depict the best architecture found by NAS methods. We conclude that NAS methods can find top architectures in both of the search spaces.

7 Conclusion

Traditionally, the progress in RL field came mostly from the development of new methods. Neural architectures of RL agents remained relatively simple when compared to computer vision applications.

In this paper, we have applied modern neural architecture search methods for optimizing the architecture of RL agents. We have evaluated ENAS [32] and SPOS [9] methods. Both of them found better architectures than manually picked by experts. We suppose that many RL application can benefit from using better neural architectures. Testing NAS methods for larger search spaces is an interesting topic for further research.

Acknowledgments. Authors are thankful to Mikhail Konobeev.

Appendix

A Search Spaces

Table 2 describes two search spaces that we used in our experiments. An abbreviation $\{1, \dots, n\}$ means that a parameter can vary from 1 to n in a search space.

Table 2. Detailed description of the search spaces.

Search Space 1					Search Space 2				
Layer	Input	Output	Kernel	Stride	Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4	Conv-1	4	32	8	4
Conv-2	32	64	1, .., 5	2	Conv-2	32	64	2, 5	1
Conv-3	64	64	1, .., 5	1	Conv-3	64	64	2, 5	1
Padding	–	121 * 64	–	–	Conv-4	64	64	2, 5	1
Linear	121 * 64	512	–	–	Conv-5	64	64	2, 5	1
					Conv-6	64	64	2, 5	1
					Padding	–	121 * 64	–	–
					Linear	121 * 64	512	–	–

B Nature CNN Architecture

Table 3 describes the architecture “Nature CNN” [18] which is a member of the search space 1.

Table 3. Convolution network architecture that used in [18] that’s contain 3 convolution layer followed by flatten and linear layers.

Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4
Conv-2	32	64	4	2
Conv-3	64	64	3	1
Padding	–	121 ∈ 64	–	–
Linear	121 ∈ 64	512	–	–

C Implementation Details

In this section, we present the hyperparameters used for training the RL agents in the ATARI games environment (Table 4), as well as the hyperparameters used for ENAS and SPOS (Table 5). We use the same set of parameters for both training from scratch experiments, and training the child networks sampled by the NAS controllers.

Table 4. The hyperparameters for training PPO agents on ATARI games.

Hyperparameter’s name	Value
# timesteps	10M
# runner timesteps	128
ϵ (PPO)	0.1
value loss coef. (PPO)	0.25
λ (GAE)	0.95
entropy coef	0.01
learning rate	CosineAnnealing(0.00025)
# parallel env	8

Table 5. The hyperparameters for ENAS training.

Hyperparameter's name	Value
# child network epochs	10
# NAS runner timesteps	3
NAS entropy coef	0.0001
NAS learning rate	CosineAnnealing(0.001)
baseline momentum	0.2

D The Best Architectures

Tables 6, 7, 8, 9 show the best architectures for each game. The architectures tend to be similar for both of the search spaces except the kernel size for the last layers.

Table 6. The best architecture for breakout extracted from search space 1 by SPOS using reward mean criteria.

Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4
Conv-2	32	64	4	2
Conv-3	64	64	5	1
Padding	–	$121 \in 64$	–	–
Linear	$121 \in 64$	512	–	–

Table 7. The best architecture for freeway extracted from search space 1 by SPOS using reward mean criteria.

Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4
Conv-2	32	64	3	2
Conv-3	64	64	2	1
Padding	–	$121 \in 64$	–	–
Linear	$121 \in 64$	512	–	–

Table 8. The best architecture for breakout extracted from search space 2 by SPOS using reward mean criteria.

Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4
Conv-2	32	64	5	1
Conv-3	64	64	5	1
Conv-4	64	64	5	1
Conv-5	64	64	2	1
Conv-6	64	64	2	1
Padding	–	$121 \in 64$	–	–
Linear	$121 \in 64$	512	–	–

Table 9. The best architecture for freeway extracted from search space 2 by SPOS using reward mean criteria.

Layer	Input	Output	Kernel	Stride
Conv-1	4	32	8	4
Conv-2	32	64	5	1
Conv-3	64	64	5	1
Conv-4	64	64	5	1
Conv-5	64	64	5	1
Conv-6	64	64	5	1
Padding	–	$121 \in 64$	–	–
Linear	$121 \in 64$	512	–	–

References

1. Bender, G.: Understanding and simplifying one-shot architecture search (2019)
2. Brockman, G., et al.: OpenAI Gym. arXiv, abs/1606.01540 (2016)
3. Cai, H., Zhu, L., Han, S.: ProxylessNAS: direct neural architecture search on target task and hardware. arXiv preprint [arXiv:1812.00332](https://arxiv.org/abs/1812.00332) (2018)
4. Chen, X., Xie, L., Wu, J., Tian, Q.: Progressive differentiable architecture search: bridging the depth gap between search and evaluation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1294–1303 (2019)
5. Dong, X., Yang, Y.: Searching for a robust neural architecture in four GPU hours. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1761–1770 (2019)
6. Dong, X., Yang, Y.: NAS-bench-102: extending the scope of reproducible neural architecture search. arXiv preprint [arXiv:2001.00326](https://arxiv.org/abs/2001.00326) (2020)
7. Espeholt, L., et al.: IMPALA: scalable distributed deep-RL with importance weighted actor-learner architectures. arXiv preprint [arXiv:1802.01561](https://arxiv.org/abs/1802.01561) (2018)

8. Gu, S., Holly, E., Lillicrap, T., Levine, S.: Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. In: 2017 IEEE International Conference on Robotics and Automation (ICRA), pp. 3389–3396. IEEE (2017)
9. Guo, Z., et al.: Single path one-shot neural architecture search with uniform sampling. arXiv preprint [arXiv:1904.00420](https://arxiv.org/abs/1904.00420) (2019)
10. Kandasamy, K., Neiswanger, W., Schneider, J., Poczos, B., Xing, E.P.: Neural architecture search with Bayesian optimisation and optimal transport. In: Advances in Neural Information Processing Systems, pp. 2016–2025 (2018)
11. Klyuchnikov, N., Trofimov, I., Artemova, E., Salnikov, M., Fedorov, M., Burnaev, E.: NAS-Bench-NLP: neural architecture search benchmark for natural language processing. arXiv preprint [arXiv:2006.07116](https://arxiv.org/abs/2006.07116) (2020)
12. Li, L., Talwalkar, A.: Random search and reproducibility for neural architecture search. arXiv preprint [arXiv:1902.07638](https://arxiv.org/abs/1902.07638) (2019)
13. Liang, H., et al.: Darts+: improved differentiable architecture search with early stopping. arXiv preprint [arXiv:1909.06035](https://arxiv.org/abs/1909.06035) (2019)
14. Liu, H., Simonyan, K., Yang, Y.: Darts: differentiable architecture search. arXiv preprint [arXiv:1806.09055](https://arxiv.org/abs/1806.09055) (2018)
15. McCandlish, S., Kaplan, J., Amodei, D., OpenAI Dota Team: An empirical model of large-batch training. arXiv preprint [arXiv:1812.06162](https://arxiv.org/abs/1812.06162) (2018)
16. Mnih, V., et al.: Asynchronous methods for deep reinforcement learning. In: International Conference on Machine Learning, pp. 1928–1937 (2016)
17. Mnih, V., et al.: Playing ATARI with deep reinforcement learning. arXiv preprint [arXiv:1312.5602](https://arxiv.org/abs/1312.5602) (2013)
18. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J.: Human-level control through deep reinforcement learning, pp. 2–3 (2015)
19. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. arXiv preprint [arXiv:1802.03268](https://arxiv.org/abs/1802.03268) (2018)
20. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 4780–4789 (2019)
21. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms (2017)
22. Shi, H., Pi, R., Xu, H., Li, Z., Kwok, J.T., Zhang, T.: Multi-objective neural architecture search via predictive network performance optimization. arXiv preprint [arXiv:1911.09336](https://arxiv.org/abs/1911.09336) (2019)
23. Silver, D., et al.: Mastering the game of go without human knowledge. Nature **550**(7676), 354–359 (2017)
24. Sutton, R.S., McAllester, D.A., Singh, S.P., Mansour, Y.: Policy gradient methods for reinforcement learning with function approximation. In: Advances in Neural Information Processing Systems, pp. 1057–1063 (2000)
25. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: a physics engine for model-based control. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 5026–5033. IEEE (2012)
26. Trofimov, I., Klyuchnikov, N., Salnikov, M., Filippov, A., Burnaev, E.: Multi-fidelity neural architecture search with knowledge distillation. arXiv preprint [arXiv:2006.08341](https://arxiv.org/abs/2006.08341) (2020)
27. White, C., Neiswanger, W., Savani, Y.: Bananas: Bayesian optimization with neural architectures for neural architecture search. arXiv preprint [arXiv:1910.11858](https://arxiv.org/abs/1910.11858) (2019)
28. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Mach. Learn. **8**, 229–256 (1992)

29. Xu, Y., et al.: PC-DARTS: partial channel connections for memory-efficient differentiable architecture search. arXiv preprint [arXiv:1907.05737](https://arxiv.org/abs/1907.05737) (2019)
30. Ying, C., Klein, A., Christiansen, E., Real, E., Murphy, K., Hutter, F.: NAS-Bench-101: towards reproducible neural architecture search. In: International Conference on Machine Learning, pp. 7105–7114 (2019)
31. Zoph, B., Le , Q.V.: Neural architecture search with reinforcement learning. arXiv preprint [arXiv:1611.01578](https://arxiv.org/abs/1611.01578) (2016)
32. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8697–8710 (2018)



Automatic Ensemble of Deep Learning Using KNN and GA Approaches

Ben Zagagy^(✉), Maya Herman, and Ofer Levi

Department of Mathematics and Computer Science, The Open University
of Israel, Ra'anana, Israel
{maya,Oferle}@openu.ac.il

Abstract. Selecting the correct deep learning architecture is a significant issue when training a new deep learning neural networks model. Even when all of other DL hyper-parameters are accurate, the selected architecture will define the final classification quality of the generated model. In our previous paper we described a unique classification methodology called ACKEM for efficient and automatic classification of data, based on an ensemble of multiple DL models and KNN input-based architecture selection. The ACKEM methodology does not restrict the classification to one specific model with one specific architecture, as a specific architecture might not fit some of the input data. The ACKEM methodology had a major constraint – it used a brute-force approach for selecting the most suitable K for its inner usage of the KNN algorithm. In this paper, we propose a genetic algorithm (GA) based approach, for selecting the most suitable K. This method was tested over multiple datasets including the Covid-19 Radiography Chest X-Ray Images Dataset, the Malaria Cells Dataset, the Road Potholes Dataset, and the Voice Commands Dataset. All the tested datasets served us in our previous work on ACKEM, as well. This paper proves that replacing the inefficient method of brute force with a GA approach can improve the ACKEM method's complexity without harming its promising results.

Keywords: Data mining · Deep learning · Ensemble classifier · KNN · Genetic algorithm · GA

1 Introduction

The “Deep Learning” method has emerged in recent years and has greatly influenced the entire Computer Science Community. Since its first appearance, deep neural networks achieved results with high quality and accuracy in many computer vision problems, even those that research teams have considered the most difficult to crack.

Correct architecture selection for a deep learning model is crucial, as emphasized in the article “Selecting the Best Architecture for Artificial Neural Networks” [3]. Selecting a good architecture can improve the classification in terms of both accuracy and time. The work “Deep Neural Network Ensembles for Time Series Classification” [4] demonstrates how ensembles of deep neural networks, can achieve state-of-the-art performance for time series classification. In our previous paper ACKEM [1], we offered

a new mechanism for DL architectures ensemble. We have shown that an ensemble of deep learning architectures could generate results with higher accuracy than single architecture deep learning models can achieve. Our solution for performing this ensemble involves searching the training input, for the nearest neighbors of the given input to be classified.

The search and neighbors finding process is performed using the well-known algorithm of K Nearest Neighbors (KNN). In the important article by Wu Et El published in 2008 [5], KNN was selected to be one of the topmost important algorithms in the field of data mining.

The models with the architectures that best classify the neighbors of a given input, were proven to best classify the given input itself. The idea was to use an optimization algorithm in order to select the best fitted deep learning model for a particular input, out of a set of given models (each based on a different architecture).

The major constraint in our previous work was the method of selecting the parameter K for the KNN phase of the solution. Even though the usage of the brute force method, yielded the best chances of finding the optimal K that will allow the system to retrieve its optimal classification results, it is still a time consuming and highly inefficient method. In this paper we propose a more efficient and less time-consuming solution for selecting the parameter K, using a GA based approach instead of the brute force approach.

Genetic Algorithms (GAs) are adaptive methods which may be used to solve search and optimization problems [6]. They are based on the genetic processes of biological organisms. Over many generations, natural populations evolve according to the principles of natural selection. By mimicking this process, genetic algorithms are able to “evolve” solutions to real world problems, provided they have been suitably encoded. GAs work with a population of individuals, each representing a possible solution to a given problem. Each individual is assigned a “fitness score” according to how good a solution to the problem it is. For example, the fitness score might be the strength/weight ratio for a given bridge design. (In nature this is equivalent to assessing how effective an organism is at competing for resources). The highly fit individuals are given opportunities to “reproduce”, by “cross breeding” with other individuals in the population. This produces new individuals as “offsprings”, which share some features taken from each of its parents. The least fit members of the population are less likely to get selected for reproduction. This GA process is repeated until a termination condition has been reached. Common terminating conditions are: minimum criteria was satisfied from the last offspring, a predefined fixed number of generations has been reached, allocated budget has been reached (could be computation time or actual money), the highest ranking available for the problem has been reached, or of course any combination of the above reasons. The pseudo code for the traditional GA is shown below in Fig. 1.

This paper is organized as follows:

- Section 2 presents the datasets used to test the ACKEM with GA method.
- Section 3 overviews our methodology for the problem solving.
- Section 4 presents the results for the case studies of an actual models ensemble using ACKEM’s novel input based KNN method with GA of deep learning models over the tested datasets.
- Section 5 provides this paper’s summary and conclusions.

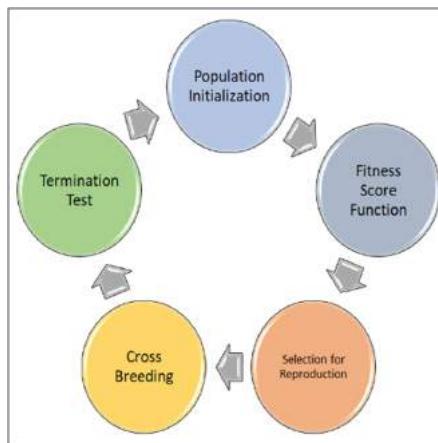


Fig. 1. Pseudo-Code for the traditional GA

2 Datasets

The considered datasets were also used in our previous work to test the ACKEM methodology with its brute-force based K selection. Testing the ACKEM's upgraded and efficient version on top of the same datasets, will ensure that the results were not harmed during transition to the more efficient version of the algorithm. The datasets that will be the base of this paper's experiments are Kaggle's Pneumonia X-Ray Dataset, Kaggle's Malaria Cell Images dataset, Kaggle's Pothole Detection Images dataset and Kaggle's Voice Commands Dataset. Further description of the used datasets is shown in Table 1.

Table 1. The number of classes, data types and amount of used training and testing data for each of the different datasets

Dataset	Number of classes	Type of data	Amount of training data	Amount of testing data
Covid19 and Pneumonia X-Ray images	2	X-Ray images	320	418
Malaria cells images	2	Cell images	2000	2000
Potholes images	2	Road images	3146	110
Voice commands	12	Audio voice commands	65,000	160,000

3 Methodology and Algorithm

As Described in our previous work, the ACKEM method consists of two core components: batch offline process for training the system and real time online classification of a given input data. The batch offline process is responsible for data retrieval and data preprocessing, it includes:

- Generation of trained deep learning models from different architectures.
- Generation of coordinates for the training data in the KNN space.
- Generation of mapping between the training data and the models that best classify them.

The online process uses all data generated by the offline process, to generate best system classification results for new inputs.

3.1 Offline Process

The offline batch process is responsible for training the Deep Learning models and generating training data coordinates in space. The generated models will be used during the online process. The Offline process contains the following four phases:

1. **Models Training.** Training models of different architectures to be used for future classification of new inputs. In order to generate such models, labeled data should be loaded into the system and different deep learning architectures should be selected for each of the generated models.
2. **Coordinates Generation.** For each of the training data inputs, coordinates in the KNN Space will be generated (aka x,y). The generated coordinates will be used in the online phase for finding nearest neighbors of new inputs, in the KNN space. Neighbors will be found according to the distances in the KNN space between the training data locations and the location of the new input. In order to generate the location of a specific input data in the KNN space, The T-SNE technique for reducing the dimensionality of large, high dimension datasets - is used. The purpose of the T-SNE technique is to reduce an input's dimensionality by reorienting it along its principal axes; it tends to preserve point-wise distances which make it suitable for visualization of high-dimensional data [7].
3. **Best Model Finder.** Generate the training data's best models dictionary, to be used during the online phase in order to decide on the best model for a specific input. In order to generate this data, there is a need to loop over all of the labeled training data, and then a classification is made using all the models generated in step #1. The model that “bests classifies” a specific training input is the one that classifies the input correctly and has the best classification accuracy, out of all models generated in step #1.
4. **K Selector using GA.** Finding the parameter K that best classifies the testing data, is done using a genetic algorithm method. The K parameter found in this phase will be used as the number of neighbors that will be selected during the Online phase.

A further explanation for phases 1–4 can be found in the ACKEM [1] paper.

3.1.1 Online Process

The purpose of this module is to find the optimal number of neighbors (K) that gives the best classifications results for the system in an efficient way.

Input: A labeled training input dataset, a non-labeled testing dataset, the matching models dictionary generated in the **Best Model Finder** phase, the training data coordinates in the KNN spaces that were generated in the **Coordinates Generator** phase and the deep learning models generated in the **Models Training** phase.

Output: The number K that generates the best system results.

Process: This module is using a genetic algorithm approach in order to find the number of neighbors with which the best results can be achieved.

This number will be referred to as K. The process of K selection, using a GA approach, is described in the pseudo code shown in Fig. 2.

3.1.2 Online Process

The Online process is based on the three types of outputs that were generated during the offline phase – the trained models, the training data mapping in the KNN space and the number of optimum neighbors for the system. The Online process is based on a new input data and the outputs from the offline process phase. For each different input for classification, the selection of a specific model (and architecture) will be performed by applying the K Nearest Neighbors algorithm; to find out which model best classifies the input's neighbors. This paper assumes that a model that correctly classifies the input's neighbors will also be able to correctly classify the input itself. The Online process contains the following three phases:

1. **Find K Nearest Neighbors.** Find the K Nearest neighbors of the new input data for classification from the training dataset in the KNN Space.
2. **Find best model for classification.** Find the model that bests classifies the K neighbors found in the previous phase.
3. **Classification.** Classify the new input data for classification, using the best fitting model, found in the previous phase.

The below pseudo code (Fig. 3) describes the sequence of events in the online phases and how the system classifies new input data:

Children Generation Algorithm Input:

- Two numbers that represent the K value of the father and mother of the child to be created.

Children Generation Algorithm:

- Set father-binary string as the binary number of the father's K value.
- Set mother-binary string as the binary number of the mother's K value.
- Set child-binary string to be an empty string.
- Loop from i=0 to i=max(father-binary length, mother-binary length):
 - If (i > father-binary length) or (i > mother-binary length) or (father-binary[i] does not equals mother-binary[i]):
 - Set the variable is-random-chromosome to True.
 - Else:
 - Set the variable is-random-chromosome to either True or False randomly.
 - If the is-random-chromosome variable is False:
 - Set the next character in the child-binary string to be mother-binary[i].
 - Else:
 - Set the next character in the child-binary string to be either 0 or 1 randomly.
- Return the decimal value of the child-binary string.

Main Algorithm Inputs:

- Total Amount of Generations = 100
- Function for calculating the system's grade using the ACKEM methodology based on a given K (calcGrade(k)).
- Maximum Value of K

Main Algorithm:

- Initialize a Current-Population array to contain 4 people, each person in this array will have a number between 1 to the maximum value of K.
- Initialize an empty "Next-Population" array.
- Set best-score variable to 0.
- Set best-K variable to 0.
- While current-generation < Total-Amount-Of-Generations:
 - For each person in the Current-Population:
 - Get the person's score using its K and the calcGrade input function.
 - If the person's score is higher than the best-score:
 - Set best-score to be the person's score.
 - Set best-K to be the person's K.
 - Add the person to the Next-Population array.
 - Else:
 - If the current person's score is higher than the Next-Population's second-best score person:
 - remove the second-best score person from the Next-Population array.
 - Add the current-person to the Next-Population array.
 - Set Current-Population array to contain 4 Children from of the two parents from the Next-Population array, each one of the children was generated using the Make Child Algorithm.
 - Set the Next-Population array to be empty.
 - Return the best-K and the best-Score found in the GA.

Fig. 2. Pseudo code for the K selection algorithm based on the genetic algorithm approach

1. Find the new Input Data coordinates in the KNN Space (aka its X, Y).
2. Place the new input Data in the KNN Space.
3. Find the new input Data's K nearest neighbors in the KNN Space using simple geometry.
4. Create an empty scores dictionary:
 - a. initialize the scores dictionary's keys with all the system's models names.
 - b. initialize the scores dictionary's values with 0 for all the models.
5. Loop through the K nearest neighbors from the training dataset, were found in step 3.
 - a. For each input data from the training dataset, retrieve the deep learning model that best classifies it, using the "Training data to their best fitting models" dictionary.
 - b. Add 1 in the scores dictionary to the value field of the model found in step 5a.
6. Retrieve the model with the highest score in the scores dictionary.
7. Using the deep learning model that was found in step 6, classify the new input data, and return its classification result.

Fig. 3. Pseudo code for the online phase's sequence of events

4 Results

4.1 Case Studies

4.1.1 Covid19 Chest X-Ray Images Dataset

The Dataset for this experiment was taken from Kaggle's "Covid-19 Radiography Database -Chest X-Ray-Dataset" [8]. This dataset contains 2 different classes of Chest X-Ray Images: Chests X-Ray images that were taken from people who are infected with Covid19 and Chests X-Ray images that were taken from people who are infected with viral Pneumonia. The data from this dataset, contains 320 labeled files, used as training data and 418 labeled files, used as testing data. The optimal results achieved using the inefficient brute force approach are shown in Fig. 4.

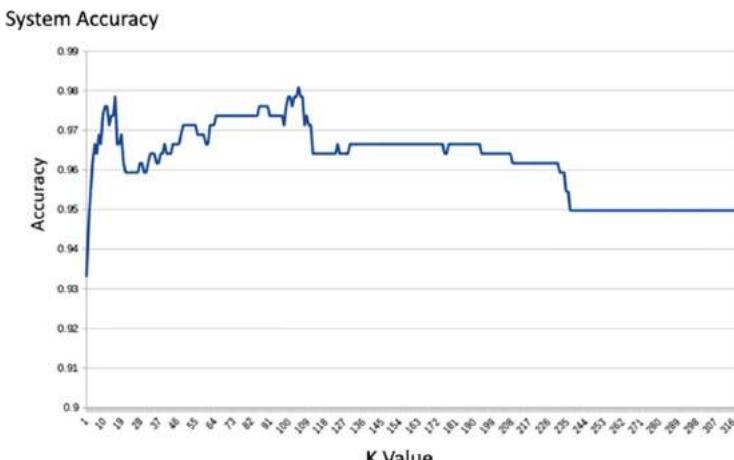


Fig. 4. Line chart describing the different values of K (X axis) against the percentage of correct classifications (Y axis) out of the Covid19 chest X-ray images testing data

As shown in Fig. 4, when choosing K to be **104**, we achieved a total amount of **98.08%** of correct classifications within the Covid19 Chest X-Ray Images testing data.

When using a GA based approach with a 100 generations termination condition, the achieved K was found to be **15** and the correct classifications grade was **97.4%**.

4.1.2 Malaria Cell Images Dataset

The Dataset for this experiment was taken from Kaggle’s “Malaria Cell Images Dataset” [9]. This dataset contains cells images of two different classes: Images of cells that were infected with Malaria and Images of cells that were not infected with Malaria. The data from this dataset, contains 2000 labeled files used as training data and 2000 labeled files used as testing data. The optimal results achieved using the inefficient brute force approaches are shown in Fig. 5.

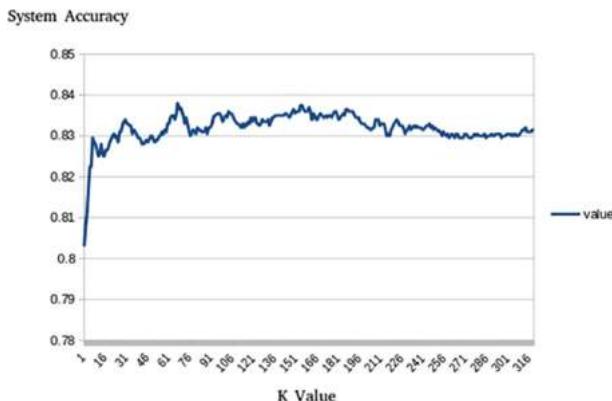


Fig. 5. Line chart describing the different values of K (X axis) against the percentage of correct classifications (Y axis) out of the malaria cells testing data

As shown in Fig. 5, when choosing K to be **67**, the total amount of **83.8%** for correct classifications within the Malaria cells testing data was achieved.

When using a GA based approach with a 100 generations termination condition, the achieved K was found to be **30** and the correct classifications grade was **83.4%**.

4.1.3 Potholes Detection Dataset

The Dataset for this experiment was taken from Kaggle’s “Pothole Detection Dataset” [10]. This dataset contains roads images of two different classes – images of roads that contain potholes that require repair and images of roads that contain no potholes and are in good shape. The data from this dataset contains 3146 labeled files used as training data and 110 labeled files used as testing data. The optimal results achieved using the inefficient brute force approaches are shown in Fig. 6.

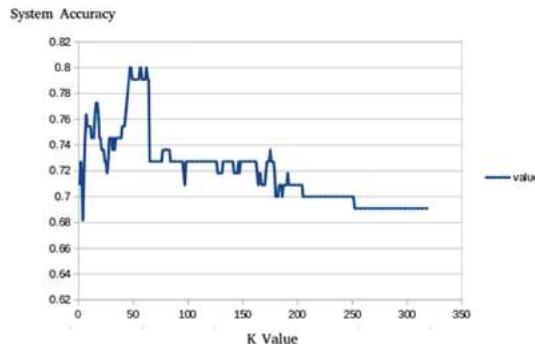


Fig. 6. Line chart describing the different values of K (X axis) against the percentage of correct classifications (Y axis) out of the roads potholes testing data.

As shown in Fig. 6, when choosing K to be **47**, the total amount of **80%** of correct classifications within the Malaria cells testing data was achieved.

When using a GA based approach with a 100 generations termination condition, the achieved K was found to be **48** and the correct classifications grade was **80%**.

4.1.4 Voice Commands Dataset

The Dataset for this experiment was taken from Kaggle’s “TensorFlow Speech Recognition Challenge” [11]. This dataset contains 12 different classes of voice commands, including: Yes, No, Up, Down, Left, Right, On, Off, Stop, Go, Unknown, Silence. This dataset includes over 65,000 labeled files for its training data and almost 160,000 files in its testing data. Firstly, in order to use (either train or test) the data from this dataset, there was a need to transform the audio files into images as we have shown in our previous work MESRS [2]. Conversion of the audio files’ content into the image space is performed. The conversion generates Mel Spectrogram images out of audio clips. An example of such image is shown in Fig. 7.

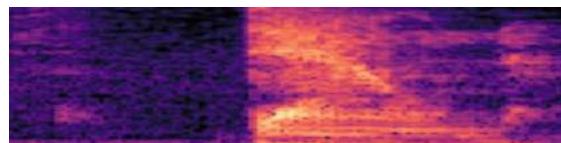


Fig. 7. An example of the mel spectrogram image generated from the word “Down”

The optimal results achieved using an inefficient brute force approaches are shown in Fig. 8.

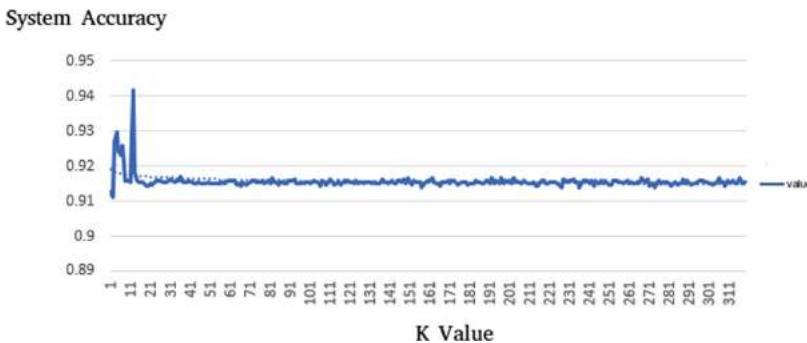


Fig. 8. Line chart describing the different values of K (X Axis) against the percentage of correct classifications (Y Axis) out of the voice commands testing data.

As shown in Fig. 8, when choosing K to be **12**, the total amount of **94.15%** of correct classifications within the Malaria cells testing data was achieved.

When using a GA based approach with a 100 generations termination condition, the achieved K was found to be **12** and the correct classifications grade was **94.15%**.

4.2 Case Studies Summary

As seen from all of the above case studies (Table 2), even though the method for selecting the parameter K has changed and it does not iterate over all the available options, the GA methodology for K selection yields almost the same results as the brute-force approach. We see that for the Malaria Cells and the Potholes Detection datasets -optimal results are achieved with the enhanced GA based approach. For the Voice Commands dataset, the GA based grade was slightly lower by 0.4% from the optimal brute-force based grade. For the COVID-19 and pneumonia dataset the GA based grade was lower by less than 1% from the optimal brute-force based grade, as well. The limitation and main disadvantage when using a GA methodology for selecting the K parameter for an ensemble, is of missing the best solution and settling for a solution that would not have been chosen

Table. 2. Results summary and comparison between the Brute-Force and GA based approaches using the case studies' datasets

Dataset	K value from Brute-Force	Grade using Brute-Force	K value from GA	Grade using GA
Covid19 vs Pneumonia	104	98.08%	15	97.4%
Malaria	67	83.8%	30	83.4%
Potholes	47	80%	48	80%
Voice commands	12	94.15%	12	94.15%

in a brute force methodology. Such cases are demonstrated in both the “Malaria” and “Covid19 vs Pneumonia” datasets.

5 Conclusions

The idea behind the ACKEM methodology of Automatic Classification based KNN Ensembling of Models - was to use an optimization algorithm for selecting the best fitted deep learning model out of a set of given models(each based on a different architecture) for a particular input. As algorithm for this selection we proposed the KNN algorithm for finding the nearest K neighbors. The uniqueness of the approach is in that, it significantly improves the conventional methods, in which one and only one architecture is selected at the beginning of the deep neural network training process. The single architecture selected might not be suitable for a specific input data, given for classification. In contrast, the ACKEM system holds a large number of architectures and, as mentioned, will select the most appropriate architecture for each specific input to be classified.

The usage of a brute force approach on top of the labeled testing data, yielded the best chances of finding the optimal K that would allow the system to achieve its optimal classification results. We knew that a more efficient and less time-consuming method for finding the optimal K will improve the ACKEM solution, as the brute force’s high complexity is one of the methodology’s main constraints. This paper proposes an improvement in the efficiency of our previous paper’s novel methodology for efficient classification, using models ensemble. As the original method for such an ensemble was based on the KNN algorithm and used a brute-force approach for finding the optimal K to be used, this paper proposes a GA based approach for finding the optimal K to be used. While the complexity of the brute force approach is of $O(N^2)$, the GA approach’s complexity is only $O(N \cdot \log(N))$.

Future work based on this paper, could offer an automatic method for creating an ACKEM-GA based system, given inputs consisting of training and testing datasets only. Such automatic systems could be implemented to perform dynamic improvement on top of new inputs, originally given for online classification. Those inputs could be automatically saved and labeled, to be later used for retraining the system’s core deep learning models.

To conclude, the proposed approach for selecting the optimal K using GA instead of a brute force-based approach, as part of the K selection phase in the ACKEM methodology, significantly improved the process complexity, without harming its powerful classification capabilities, as shown on top of the case studies datasets.

References

1. Zagagy, B., Herman, M., Levi, O.: ACKEM – automatic classification, using KNN based ensemble modeling. In: The Future of Information and Communication Conference (2021)
2. Zagagy, B., Herman, M.: MESRS: models ensemble speech recognition system. In: Arai, K., Kapoor, S., Bhatia, R. (eds.) SAI 2020. AISC, vol. 1229, pp. 214–231. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52246-9_15
3. Ahmed, G.: Selecting the Best Architecture for Artificial Neural Networks (2019)

4. Fawaz, H.I., et al.: Deep neural network ensembles for time series classification. In: 2019 International Joint Conference on Neural Networks (IJCNN). IEEE (2019)
5. Wu, et al.: Top 10 algorithms in data mining (2008)
6. Yadav, P.K., Prajapati, N.L.: An overview of genetic algorithm and modeling. *Int. J. Sci. Res. Publ.* **2**(9), 1–4 (2012)
7. van der Maaten, L., Hinton, G.: Visualizing data using t-SNE. *J. Mach. Learn. Res.* **9** (2008)
8. Kaggle's Covid-19 Radiography Database - Chest X-Ray-Dataset
9. Kaggle's Malaria Cell Images Dataset
10. Kaggle's Potholes Detection Dataset
11. Kaggle's TensorFlow Speech Recognition Voice Commands Dataset



A Deep Learning Model for Data Synopses Management in Pervasive Computing Applications

Panagiotis Fountas¹(✉), Kostas Kolomvatsos¹, and Christos Anagnostopoulos²

¹ Department of Computer Science and Telecommunications, University of Thessaly, Papasiopoulou 2-4, 35131 Lamia, Greece
{pfountas,kostasks}@uth.gr

² School of Computing Science, University of Glasgow, Lilybank Gardens 17, Glasgow G12 8RZ, UK
christos.anagnostopoulos@glasgow.ac.uk

Abstract. Pervasive computing involves the placement of processing units and services close to end users to support intelligent applications that will facilitate their activities. With the advent of the Internet of Things (IoT) and the Edge Computing (EC), one can find more room for placing services at various points in the interconnection of the aforementioned infrastructures. Of significant importance is the processing of the collected data to provide analytics and knowledge. Such processing can be realized upon the EC nodes that exhibit increased computational capabilities compared to IoT devices. An ecosystem of intelligent nodes is created at the EC giving the opportunity to support cooperative models towards the provision of the desired analytics. Nodes become the hosts of geo-distributed datasets formulated by the reports of IoT devices. Upon the datasets, a number of queries/tasks can be executed either locally or remotely. Queries/tasks can be offloaded for performance reasons to deliver the most appropriate response. However, an offloading action should be carefully designed being always aligned with the data present to the hosting node. In this paper, we present a model to support the cooperative aspect in the EC infrastructure. We argue on the delivery of data synopses distributed in the ecosystem of EC nodes making them capable to take offloading decisions fully aligned with data present at every peer. Nodes exchange their data synopses to inform their peers. We propose a scheme that detects the appropriate time to distribute the calculated synopsis trying to avoid the network overloading especially when synopses are frequently extracted due to the high rates at which IoT devices report data to EC nodes. Our approach involves a deep learning model for learning the distribution of calculated synopses and estimate future trends. Upon these trends, we are able to find the appropriate time to deliver synopses to peer nodes. We provide the description of the proposed mechanism and evaluate it based on real datasets. An extensive experimentation upon various scenarios reveals the pros and cons of the approach by giving numerical results.

Keywords: Edge computing · Internet of Things · Data management · Data synopsis · Deep learning

1 Introduction

Pervasive computing targets to the creation of smart environments around end users saturated with computing and communication capabilities to support novel applications. Pervasive services aim to be invisible, however, intelligent enough to facilitate users activities. Today, we are witnessing the provision of huge infrastructures where pervasive applications can be hosted. Such infrastructures deal with the Internet of Things (IoT) and Edge Computing (EC). Both of them try to “surround” end users with smart devices, collect and process data to create knowledge adopted by various applications. Obviously, in this new era of the Web, there are numerous opportunities to support intelligent and invisible services in a close distance with users. Hence, we are able to reduce the latency in the provision of the discussed services and increase the performance. The first “actor” in this setting is the IoT device that may directly interact with users and their environment to collect data and perform simple processing activities [38]. IoT devices can, then, report their data in an upwards mode, to the EC infrastructure and Cloud for further processing. EC involves an ecosystem of heterogeneous nodes that become the hosts of the collected data and act as processing points to deliver analytics and knowledge [38]. We can observe a high number of distributed datasets present at the network edge opening the room for defining advanced services and support real time applications. The aim is to serve users or applications requests in the minimum time with the maximum performance. As the maximum performance we denote the provision of responses that fully match to the incoming requests. Obviously, responses are provided upon the available data and should be aligned with them.

As the EC supports a distributed environment with numerous datasets present at the ecosystem of nodes, the use of cooperative models is imposed to make nodes capable of exchanging data, queries/tasks, etc. The interaction between EC nodes aims at detecting the appropriate line of actions to efficiently respond to the incoming requests for processing. The research community has already focused not only on the management of queries/tasks [21, 23, 24, 26, 27] but also on the management of the collected data [5]. However, due to the distributed nature of the EC, nodes should have a view on the data present in peers especially when we want to support efficient decision making locally [6]. For instance, a data allocation action demands for the knowledge of the remote data at least in the form of synopses. Data synopses can be exchanged between EC nodes without burdening the network as they usually convey high level statistical information about the available data [22]. The delivery of synopses seems to be more efficient than the exchange of large pieces of data, i.e., data migration between nodes [10]. Actually, we have two solutions for responding to queries/tasks requests when the relevant data are not present at the node receiving the request. The first solution deals with the migration of queries/tasks

upon the decision of finding the most appropriate node as concluded by the corresponding synopsis. The second solution deals with the migration of the relevant data from the appropriate owner/peer to the node receiving the request. Evidently, the former model should be supported by an intelligent mechanism for exchanging the necessary statistical information for the data present in the ecosystem while the latter scheme burdens the network with large messages increasing the possibility of bottlenecks.

In this paper, we support the autonomous nature and the cooperation between EC nodes to serve queries/tasks demanded by users or applications. We focus on the first of the aforementioned solutions (i.e., queries/tasks migration) and propose a scheme for exchanging data synopses in the ecosystem to efficiently support decisions related to offloading actions. Synopses are updated every time new data arrive in an EC node, however, they should be distributed when their “magnitude” exhibit that a significant new information is present. We propose a monitoring mechanism for the updated synopses and a model that detects when significant changes are present at every dataset. When this is true, we decide to deliver the synopses to peer nodes to have them informed about the new status of every dataset. We rely on a Deep Machine Learning (DML) model to learn the distribution behind the calculated synopses being able to estimate their future trends. Hence, in a proactive manner, we are able to estimate the appropriate time for delivering the updated synopses. More specifically, we adopt a Long Short Term Memory (LSTM) network which is a specific type of Recurrent Neural Networks (RNNs) [19]. The adopted LSTM is capable of incorporating data from the previous step to the upcoming steps of processing. Hence, LSTMs are capable of identifying dependencies on data that “legacy” neural networks cannot do. The detection of such dependencies are critical in our scenario as synopses are updated in an “incremental” manner, i.e., new data arrivals are affecting the statistical information of datasets that is related to the previously delivered synopses. We consider the trade off between the frequency of synopses distribution and the “magnitude” of updates. We can accept the limited freshness of updates for gaining benefits in the performance of the network. We also define an uncertainty driven model under the principles of Fuzzy Logic (FL) [39] to decide when an EC node should distribute the synopsis of its dataset. The uncertainty is related to the “threshold” (upon the differences of the available data after getting reports from IoT devices) over which the node should disseminate the current synopsis. We monitor the “statistical significance” of synopses updates before we decide to distribute them in the network. We consider the trade off between the frequency of the distribution and the “magnitude” of updates. We can accept the limited freshness of updates for gaining benefits in the performance of the network. We apply our scheme upon past, historical observations (i.e., synopses updates) as well as upon future estimations. Both, the view on the past and the view on the future (derived by the proposed LSTM) are fed into our Type-2 FL System (T2FLS) to retrieve the *Degree of Distribution* (DoD). Two DoD values (upon historical values and future estimations) are smoothly aggregated through a geometrical mean function [36]

to finally decide the dissemination action. Our contributions are summarized by the following list:

- We support monitoring activities for detecting the magnitude of the updated synapses.
- We deliver an LSTM for learning the distribution and dependencies on continuous updates of data synapses for estimating their future realizations.
- We propose an uncertainty driven model for detecting the appropriate time to distribute data synapses to peers.
- We report on the experimental evaluation of the proposed models through a large set of simulations.

The paper is organized as follows: Sect. 2 presents the related work while Sect. 3 formulates our problem and provides the main notations adopted in our model. In Sect. 4, we present the envisioned mechanism and explain its functionalities. In Sect. 5, we deliver our experimental evaluation and conclude the paper in Sect. 6 by presenting our future research directions.

2 Related Work

Resource management at the EC has been already studied in the past to reveal the requirements for hosting and processing data. This is because data processing demands for specific resources according to the complexity of the requested queries/tasks. The appropriate allocation of the available resources will guarantee the increased performance and the timely provision of the outcomes [25]. A number of efforts try to deal with the resource management problem [7, 14, 43, 47]. Their aim is to address the challenges on how we can offload various tasks/queries and data to EC nodes taking into consideration a set of constraints, e.g. time requirements, communication needs, nodes' performance, the quality of the provided responses and so on and so forth. A relevant study in the domain reveals that processing nodes may adopt the following three (3) schemes to perform the execution of queries/tasks [47]:

- An integration model for aggregating data reported by multiple devices [48]. EC nodes have the opportunity to locally process the collected data before they transfer them to the Cloud. This approach limits the time for the provision of the final response as the processing is performed in close distance with end users.
- A “cooperative” scheme through which EC nodes interact with other devices (e.g. IoT devices, EC nodes) having processing capabilities to offload a subset of tasks [49]. In any case, the distance between the interacting devices should be low, otherwise, their interaction may be problematic.
- A “centralized” approach where EC nodes act as execution points for queries/tasks offloaded by IoT devices [40]. This approach sees EC nodes having increased computational resources compared to IoT devices, thus, they can perform more complicated processing. Arguably, EC nodes should incorporated a monitoring mechanism to detect possible overloading cases and take specific mitigation actions.

Additionally, for speeding up the processing at the EC nodes while being aligned with the requirements of requests, various efforts have proposed the use of caching [11], context-aware web browsing [41], and pre-processing actions [44].

Evidently, queries/tasks are executed upon huge volumes of data. The processing of large scale data demands for efficient techniques to timely deliver the outcomes. The support of synopses management is already identified by the research community as a means for having a view upon the data avoiding to perform time consuming activities. Synopses convey statistical information about the underlying data [3] and can be maintained in an incremental approach. The research community has connected the term “synopsis” with (i) approximate query estimation [12]: the target is to estimate, in real time, responses given the query and without having any view on data. Obviously, the final goal is to detect the data that better “match” to the incoming queries; (ii) approximate join estimation [4,17]: join operations are usually time consuming and more complex compared to other types of operations (e.g., a simple select upon the available data). Hence, approximate solutions may limit the time required to conclude any join operation taking into consideration the trade off between the accuracy of results and the conclusion time; (iii) aggregates calculation [13,16,18,32]: the aim is to provide aggregate statistics over the available data; (iv) data mining schemes [1,2,42]: there are services demanding for synopses instead of the individual data points, e.g., clustering, classification. This means that any decision is retrieved upon the high level statistical information for the available data. In any case, the adoption of data synopses aims at the processing of only a subset of the actual data. Synopses act as “representatives” of data and usually involve summarizations or the selection of a specific subset [31]. Any limited representation may heavily reduce the need for increased bandwidth of the network and can be easily transferred in the minimum possible time. Example techniques for the delivery of synopses deal with sampling [31], load shedding [8,45], sketching [9,37] and micro cluster based summarization [1]. The easiest technique is sampling. It targets to the probabilistic selection of a subset of the actual data. Obviously, the appropriate selection of samples plays a significant role in the success of the decision making model for which the selected samples become the subject of processing. Load shedding aims to drop some data when the system identifies a high load, thus, to avoid bottlenecks. Sketching involves the random projection of a subset of features that describe the data incorporating mechanisms for the vertical sampling of the stream. Micro clustering targets to the management of the multi-dimensional aspect of any data stream towards to the processing of the data evolution over time. Other statistical techniques are histograms and wavelets [3].

3 Preliminaries and Problem Description

We focus on the ecosystem of EC nodes that exhibit cooperative behaviour towards the execution of the received queries/tasks. Without loss of generality, we assume N EC nodes depicted by the set $\mathcal{N} = \{n_1, n_2, \dots, n_N\}$. Every node

hosts the corresponding dataset, thus, N geo-distributed datasets are available as depicted by the following set $\mathcal{D} = \{D_1, D_2, \dots, D_N\}$. Datasets contain multivariate vectors reported by the IoT devices being connected with EC nodes. n_i hosts the reports of its IoT devices and formulates a dataset $D_i = \{\mathbf{x}_j\}_{j=1}^{m_j}$ with m_j real-valued contextual multidimensional data vectors. Each data vector $\mathbf{x} = [x_1, x_2, \dots, x_d]' \in \mathbb{R}^d$ involves data related to d dimensions. For instance, IoT devices may monitor a phenomenon and report sensory data related to it (e.g., they could monitor a fire event and report data for temperature, humidity, etc.). Any processing activity in n_i is performed upon D_i and targets to the provision of analytics or knowledge. For instance, requests may demand for a regression analysis, classification, the estimation of multivariate and/or uni-variate histograms per attribute, non-linear statistical dependencies between input attributes and an application-defined output attribute, clustering of the contextual vectors, etc. D_i s are also the basis for delivering the discussed synopses. Let us denote a statistical synopsis by \mathbf{s} . \mathbf{s} is depicted by l -dimensional vectors, i.e., $\mathbf{s} = [s_1, s_2, \dots, s_l]' \subset \mathbb{R}^l$. As mentioned above \mathbf{s}_i delivered by n_i is the summarization of D_i upon the data reported by the corresponding IoT devices [29]. Obviously, there are N synopses $\mathbf{s}_1, \dots, \mathbf{s}_N$ that have to be distributed in the ecosystem of EC nodes.

Let us focus on the behaviour of a specific EC node n_i . A similar approach dictates the behaviour of all the available nodes in the EC ecosystem. Initially, n_i is responsible to calculate locally the corresponding synopsis \mathbf{s}_i upon D_i . This happens when the received data change the statistics of the underlying dataset (e.g., concept drift). Afterwards, n_i tries to act in a cooperative manner and decide to exchange \mathbf{s}_i regularly. The target is to inform peers about the changes in the statistics of its dataset, thus, to give them the opportunity to be aligned with new trends in D_i . Obviously, n_i , before sending \mathbf{s}_i , should take into consideration the trade off between the communication overhead and the “freshness” of \mathbf{s}_i delivered to peers. n_i can share up-to-date synopses every time a change (even the smallest one) in the underlying data is realized at the expense of flooding the network with numerous messages. We have to keep in mind that the connection of the IoT and EC infrastructures involves numerous devices exchanging numerous messages to convey data, synopses, knowledge, etc. Hence, the frequency of the delivery of messages plays a significant role in the performance of the network. In any case, a frequent delivery of \mathbf{s}_i will give the opportunity to peer nodes to enjoy fresh information increasing the performance of decision making. An intermediate solution is to postpone the delivery of \mathbf{s}_i and reduce the sharing rate expecting less network overhead in light of “obsolete” synopses. The delay in delivering \mathbf{s}_i can be dictated by the limited updates in \mathbf{s}_i as the result of retrieving data that cannot significantly change the statistics of the D_i . In this paper, we rely on the second approach and propose a model that monitors \mathbf{s}_i and detects where significant changes in the underlying data are present before it decides to deliver the updated \mathbf{s}_i . The target is to optimally limit the messaging overhead. Our rationale is to monitor the “magnitude” of the collected statistical synopsis before we decide a dissemination action. In this approach, there are two

main problems. The first is related to if past observations are the appropriate basis for initiating the delivery of \mathbf{s}_i while the second deals with the uncertainty in the adopted threshold that will “fire” the delivery action. Thresholds are set into any decision making mechanism that tries to detect the appropriate time to initiate an action. For the first problem, we proposed the use of an LSTM/RNN capable of learning the dependencies of data, thus it will be easy to retrieve their future estimates. For the second problem, we proposed the use of a T2FLS to handle the incorporated uncertainty, i.e. our T2FLS results the DoD upon past synopsis observations and its estimated values. The proposed T2FLS tries to bridge the “gap” between past observations and future trends of synopses. In any case, EC nodes are forced to disseminate synopses at pre-defined intervals even if no delivery decision is the outcome from our model. We have to notice that, to avoid bottlenecks in the network, we consider the pre-defined intervals to differ among the group of EC nodes. This “simulates” a load balancing approach avoiding to have too many EC nodes disseminating their synopses at the same time, thus, burdening the network.

Our LSTM/RNN and T2FLS are fed by the most recent \mathbf{s}_i realizations. The LSTM/RNN retrieves future estimates of \mathbf{s}_i that are also fed into the proposed T2FLS. To the best of our knowledge, the proposed model is one of the first attempts that combines a DML with an FL system to deliver a powerful decision making mechanism. The LSTM/RNN undertakes the responsibility of learning the data and their dependencies through time and the T2FLS focuses on the management of uncertainty in decision making. n_i monitors significant changes in \mathbf{s}_i as more contextual data are received from IoT devices. Based on the local monitoring activity, implicitly, we incorporate into the network edge the necessary “randomness” in the conclusion of the final decision, thus, potentially avoiding network flooding. The discussed “randomness” is enhanced by different data arriving to the available nodes and their autonomous decision making. Such “randomness” can assist in limiting the possibility of deciding the delivery of synopses at the same time, thus, we can limit the possibility of overloading the network. We consider that at t (a discrete time instance) a new \mathbf{x} arrives in n_i . Afterwards, the corresponding synopsis \mathbf{s}_i^{t-1} should be updated to conclude the new \mathbf{s}_i^t . Let \mathbf{e}_t be the difference over the current, last sent synopsis \mathbf{s}_i^{t-1} and the new, the updated one, \mathbf{s}_i^t . We define this error as the *update quantum*, i.e. the magnitude of the difference between \mathbf{s}_i^{t-1} and \mathbf{s}_i^t . n_i calculates \mathbf{e}_t at consecutive time steps and, in a simplistic way, can be concluded by adopting the *sum of differences* between two consecutive synopsis for every dimension. In any case, we can rely on any desired synopsis realization technique. \mathbf{e}_t may have a positive or a negative trend, i.e. the new vector can increase or decrease the value of each dimension. For easiness in our calculations, we take into consideration the absolute value of any difference into the available dimensions. EC nodes should delay the delivery of \mathbf{s}_i^t until they see that a significant difference, i.e. a high *magnitude* depicted by \mathbf{e}_t is present. At that time, it is necessary to have the peer nodes informed about the new status of the local dataset. We define the *update epoch* as the time between disseminating \mathbf{s}_i^{t-1} and \mathbf{s}_i^t . The update epoch

is realized at pre-defined intervals, $T, 2T, 3T, \dots$ ($T > 0$). In this description, we focus on a single interval, e.g., $[1, 2, \dots, T]$ where EC nodes examine the last \mathbf{e} realizations and feed them into our LSTM/RNN and T2FLS to see if they excuse the initiation of the dissemination process. For sure, the dissemination of \mathbf{s}_i^t will be concluded at T if no relevant decision is made by our scheme. n_i also ‘reasons’ over the time series of update quanta $\{\mathbf{e}_t\}$ with $t = 1, 2, \dots, T$. It ‘projects’ the time series to the future through the adoption of our LSTM/RNN. Again, the projection of update quanta is fed into the T2FLS to generate the DoD upon the future estimations of \mathbf{e} .

4 Uncertainty Driven Proactive Synopses Dissemination

4.1 Estimating Future Trends of Synopses

The proposed LSTM/RNN adopts the time series of $\{\mathbf{e}_t\}$ realizations as new data vectors arrive locally. RNNs produce an output at every time step and have recurrent connections between hidden units. The output is not discrete as the LSTM/RNN is adopted to predict the future \mathbf{e}_t realizations. In general, for RNNs, the following update equations are adopted to transfer data inside the network:

$$\alpha_t = b + Wh^{t-1} + U\mathbf{e}_t \quad (1)$$

$$h^t = \tanh(\alpha_t) \quad (2)$$

$$o^t = c + Vh^t \quad (3)$$

$$\hat{y}^t = \text{softmax}(o^t) \quad (4)$$

where B and c are bias vectors, U , V and W are weight matrices for the input-hidden, hidden-output and hidden-hidden connections of the neural network. The proposed LSTM/RNN maps the input with the output that are of the same size.

We select to adopt an LSTM [19], i.e., a specific type of RNNs to capture synopses trends for each dataset. Our LSTM tries to ‘understand’ every synopsis realization based on previous realizations and efficiently learn their distribution. Legacy neural networks cannot perform well in cases where we want to capture the trend of a time series. RNNs and LSTMs are network with loops inside of them making data to persist. We have to notice that the LSTM delivers DoD_f for each synopsis realization. In our model, we adopt an LSTM for the following reasons: (i) we want to give the opportunity to the proposed model to learn over large sequences of data ($T \gg 1$) and not only over recent data. Typical RNNs suffer from short-term memory and may leave significant information from the beginning of the sequence making difficult the transfer of information from early

steps to the later ones; (ii) typical RNNs also suffer from the *vanishing gradient problem*, i.e., when a gradient becomes very low during back propagation, the network stops to learn; (iii) LSTMs perform better the processing of data compared to other architectures as they incorporate multiple ‘gates’ adopted to regulate the flow of the information. Hence, they can learn better than other models upon time series.

Every LSTM cell in the architecture of the network has an internal recurrence (i.e., a self-loop) in addition to the external recurrence of typical RNNs (see Fig. 1). It also has more parameters than an RNN and the aforementioned gates to control the flow of data. The self-loop weight is controller by the so-called forget gate, i.e., $g_f^t = \sigma(b^f + \sum_j U_j^f \mathbf{e}_j^t + \sum_j Z_j^f h_j^{t-1})$ where σ is the standard deviation of the unit, b^f represents the bias of the unit, U^f represents the input weights, \mathbf{e} is the vector of inputs (we can get as many inputs as we want out of W recordings), Z^f represents the weights of the forget gate and h^{t-1} represents the current hidden layer vector. The internal state of an LSTM cell is updated as follows: $s^t = g_f^t s^{t-1} + g_{in}^t \sigma(b + \sum_j U_j \mathbf{e}_j^t + \sum_j Z_j h_j^{t-1})$. Now, b , U and Z represent the bias, input weights and recurrent weights of the cell and g_{in} depicts the external input gate. We perform similar calculations for the external input g_{in} and the output gates g_{out} . The following equations hold true:

$$g_{in}^t = \sigma \left(b^{in} + \sum_j U_j^{in} \mathbf{e}_j^t + \sum_j Z_j^{in} h_j^{t-1} \right) \quad (5)$$

$$g_{out}^t = \sigma \left(b^{out} + \sum_j U_j^{out} \mathbf{e}_j^t + \sum_j Z_j^{out} h_j^{t-1} \right) \quad (6)$$

The output of the cell is calculated as follows:

$$h^t = \tanh(s^t) g_{out}^t \quad (7)$$

In our scenario, we adopt a multiple input, multiple output LSTM to get the future estimation of \mathbf{e} . We consider that the number of inputs/outputs are the three most recent synopsis error observations, i.e., $\mathbf{e}_{t-2}, \mathbf{e}_{t-1}, \mathbf{e}_t$ for inputs and $\mathbf{e}_{t+1}, \mathbf{e}_{t+2}, \mathbf{e}_{t+3}$ for outputs. It should be noticed that our LSTM is trained upon real datasets by calculating the synopses of the reports as we reveal in our experimental evaluation section. Past observations $\mathbf{e}_{t-2}, \mathbf{e}_{t-1}, \mathbf{e}_t$ are fed into the proposed T2FLS to retrieve the DoD_p as well as future estimations $\mathbf{e}_{t+1}, \mathbf{e}_{t+2}, \mathbf{e}_{t+3}$ are adopted by our T2FLS to retrieve the DoD_f . Hence, our decision making model delivers the appropriate outcomes based on both approaches upon the statistical information of the local synopses.

4.2 The Uncertainty Driven Model

For describing the proposed T2FLS, we borrow the notation of our previous efforts (in other domains) presented in [28,30]. T2FLS is adopted locally at

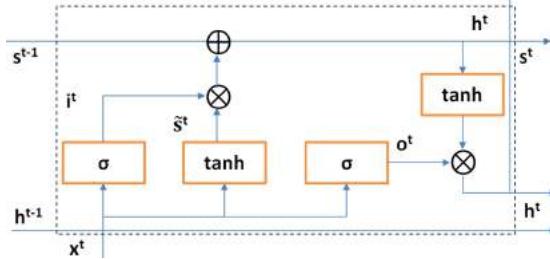


Fig. 1. The architecture of a LSTM neuron

every node at t by fusing the past \mathbf{e}_t observations and future \mathbf{e}_t realizations. \mathbf{e}_t is adopted as the indication whether the current update quanta significantly deviate from their past and future short-term trends. The envisioned fusion of update quanta is achieved through a finite set of *Fuzzy Rules* (FRs). FRs incorporate past quanta or future estimations (two different processes) to reflect the *DoD*. Actually, we ‘fire’ in two consecutive iterations the T2FLS for the last three (3) quanta realizations, i.e., $\mathbf{e}_{t-2}, \mathbf{e}_{t-1}, \mathbf{e}_t$ and the future three (3) quanta estimations, i.e., $\mathbf{e}_{t+1}, \mathbf{e}_{t+2}, \mathbf{e}_{t+3}$. Our T2FLS, defines the fuzzy knowledge base for every n_i , e.g., a set of FRs like: ‘when the past/future quanta exhibit a significant difference from the last synopsis delivery, the DoD for initiating the delivery of the new synopsis might be also high’. We rely on Type-2 FL sets as the ‘typical’ Type-1 fuzzy sets and the FRs defined upon them involve uncertainty due to partial knowledge in representing the output of the inference [35]. The limitation of a Type-1 FL system is on handling uncertainty in representing knowledge through FRs [20, 35]. In such cases, uncertainty is observed not only in the environment, e.g., we classify the *DoD* as ‘low’ or ‘high’, but also on the description of the term, e.g., ‘low’/‘high’, itself. In a T2FLS, membership functions are themselves ‘fuzzy’, which leads to the definition of FRs incorporating such uncertainty [35].

FRs refer to a non-linear mapping between three inputs: (i) when focusing on the past quanta, we take as the following as inputs into the T2FLS: $\mathbf{e}_{t-2}, \mathbf{e}_{t-1}, \mathbf{e}_t$; (ii) when focusing on future quanta, we take the following as inputs into the T2FLS: $\mathbf{e}_{t+1}, \mathbf{e}_{t+2}, \mathbf{e}_{t+3}$. The outputs are DoD_p & DoD_f , respectively. The antecedent part of FRs is a (fuzzy) conjunction of inputs and the consequent part of the FRs is the *DoD* indicating the belief that an event *actually* occurs. The proposed FRs have the following structure:

IF \mathbf{e}_{t-2} is A_{1k} AND \mathbf{e}_{t-1} is A_{2k} AND \mathbf{e}_t is A_{3k}
THEN DoD_p is B_k ,
 IF \mathbf{e}_{t+1} is A_{1k} AND \mathbf{e}_{t+2} is A_{2k} AND \mathbf{e}_{t+3} is A_{3k}
THEN DoD_f is B_k ,

where A_{1k}, A_{2k}, A_{3k} and B_k are membership functions for the k -th FR mapping $\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k$ and DoD_v , $i \in \{t-2, t+1\}$, $j \in \{t-1, t+2\}$, $k \in \{t, t+3\}$ and $v \in \{p, f\}$. For FL sets, we characterize their values through the terms:

low, *medium*, and *high*. The structure of FRs in the proposed T2FLS involve linguistic terms, e.g., *high*, represented by two membership functions, i.e., the *lower* and the *upper* bounds [34]. For instance, the term ‘*high*’ whose membership for x is a number $g(x)$, is represented by two membership functions defining the interval $[g_L(x), g_U(x)]$. This interval corresponds to a lower and an upper membership function g_L and g_U , respectively (e.g., the membership of $x = 0.25$ can be in the interval $[0.05, 0.2]$). The interval areas $[g_L(x_j), g_U(x_j)]$ for each x_j reflect the uncertainty in defining the term, e.g., ‘*high*’, useful to determine the exact membership function for each term. Obviously, if $g_L(x) = g_U(x), \forall x$, we obtain a FR in a Type-1 FL system. The interested reader could refer to [34] for information on reasoning under Type-2 FRs. We have to notice that FRs and membership functions for the proposed T2FLS are defined by experts. We consider a knowledge base with 27 rules, i.e., a rule for every combination of the three inputs.

4.3 Synopses Update and Delivery

As mentioned, in an iterative manner, our T2FLS is fed by past realizations and future estimations of the update quanta calculated upon the available datasets. The outcomes are depicted by DoD_p and DoD_f . We have to combine these two results into the final DoD that exhibits the potential of initiating the delivery of the current synopsis. In other words, we have to combine our view on the past with our estimations of the future before we decide to distribute the updated synopsis to peer nodes. For the aggregation process, we strategically select to rely on a simple methodology that will derive the final outcome in real time. We propose the use of the geometric mean [36] as the function for integrating the two aforementioned views on the updated synopses. The following equation holds true:

$$G(DoD_p, DoD_f) = \left(\prod_{i=1}^2 DoD_i \right)^{1/2} \quad (8)$$

with $i \in \{p, f\}$. The rationale behind the adoption of the geometric mean is that it is not affected by extreme values (high or low) and deals with all the inputs. Moreover, the multiplicative approach supported by the geometric mean makes our model to be ‘strict’ approach. For instance, when one out of the two DoD values is zero, the final outcome is zero as well. This way, we want to be sure that there is ‘critical’ amount of magnitude in synopses before they are distributed in the network. The final decision depends on a threshold θ . When $G > \theta$, we initiate the dissemination action. θ is a pre-defined threshold that ‘dictates’ when an EC node should pursue the exchange of a synopsis.

5 Experimental Setup and Evaluation

5.1 Setup and Performance Metrics

We report on the performance of our *Uncertainty Driven Synopses Dissemination Model* (UDSDM) and compare it with other baseline models and schemes

proposed in the relevant literature. Initially, we focus on the percentage of T that our model spends till the final decision. This is exposed by the ϕ metric which is defined as follows:

$$\phi = \frac{1}{E} \sum \left\{ \frac{t^{\leq}}{T} \right\}_{i=1}^E \quad (9)$$

where t^{\leq} is the time when the dissemination actions is decided, E is the number of experiments and i depicts the index of every experiment. When $\phi \rightarrow 1$ means that the proposed model spends the entire interval T to conclude a final decision. When $\phi \rightarrow 0$, our model manages to conclude immediately the dissemination action. Additionally, we define the metric δ i.e.,

$$\delta = \frac{1}{E} \sum \left\{ |\mathbf{s}^{t^*} - \mathbf{s}| \right\}_{i=1}^E \quad (10)$$

δ represents the average magnitude of the difference between the current and the new synapses. Through δ , we want to depict the ability of the proposed model to ‘react’ even in limited changes in the updated synapses (we target a $\delta \rightarrow 0$). The magnitude is calculated at t^{\leq} . The ability of the proposed system to avoid the overloading of the network and limiting the required number of messages is exposed by ψ . The following equation holds true ($\psi \in [0, T]$):

$$\psi = \frac{T}{|t^{\leq}_{t^* \in [1, T]}|} \quad (11)$$

where $|t^{\leq}_{t^* \in [1, T]}|$ represents the number of times that the model stops in the interval $[1, T]$. When $\psi \rightarrow 1$ means that the proposed model stops frequently, thus, multiple messages conveying the calculated synapses are transferred through the network. When $\psi \rightarrow T$ means that our model does not stop frequently, thus, the calculated synapses are delivered close to the expiration of T . For our experimentation, we adopt the dataset presented in Intel Berkeley Research Lab [15]. It contains measurements from 54 sensors deployed in a lab. We get the available measurements and simulate the provision of context vectors to calculate the synapses and the update quanta (they are realized in the interval $[0, \infty]$) in a sequential order. Upon these quanta and their distribution, we perform the training of the proposed LSTM. Based on the statistics of quanta, we produce a high number of training tuples and feed the into the LSTM. We also pursue a comparative assessment for the UDSDM with: **(i)** a Baseline Model (BM) that disseminates synapses when any change is observed over the incoming data; **(ii)** the Prediction based Model (PM) [33]: PM proceeds with the stopping decision only when the estimation of the future update quanta violates a threshold. For realizing the PM, we adopt the double exponential smoothing method [46]. The method applies a recursive model of an exponential filter twice before it results the final outcome. We perform simulations for $W = 10$ adopted to realize the double exponential smoothing scheme and $T \in \{100, 500, 1000\}$. In every experiment, we run the UDSDM and get numerical results related to the mean of the aforementioned metrics (we adopt $\theta \in \{0.60, 0.75\}$ for the UDSDM and the PM).

5.2 Performance Assessment

In Fig. 2, we present our results for the ϕ metric. In these results, we omit the BM as it is a model that reports the calculated synapses every time a new vector arrives at an EC node. We observe that the adoption of a low θ (threshold for deciding the dissemination action) leads to an decreased time for the final decision. This means that the proposed model manages to conclude immediately a fuzzy result upon θ that ‘fires’ the dissemination action. In addition, and increased T leads to a decreased ϕ . The higher the T is, the lower the ϕ becomes. When θ and T are high, the percentage of T devoted to conclude the dissemination decision is very low. Compared to the PM, the UDSDM requires less time to conclude the delivery action (except when $\theta = 0.75$ & $T = 500$) for the majority of the experimental scenarios. Actually, the proposed system manages to deal with the final decision as soon as it detects that update quanta are aggregated over time even in small amounts. This can be realized in early monitoring rounds due to the dynamic nature of the incoming data. Recall that we adopt a time series that consists of sensory data retrieved by a high number of devices that are, generally, characterized by their dynamic nature.

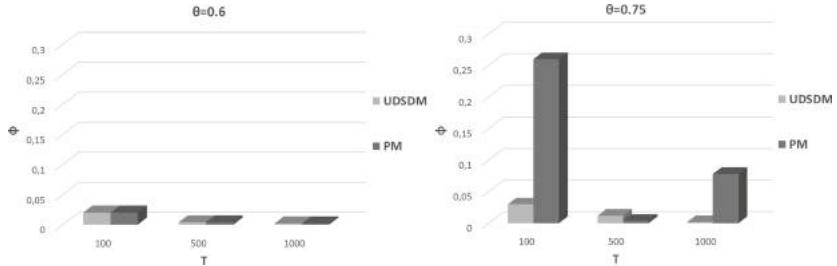
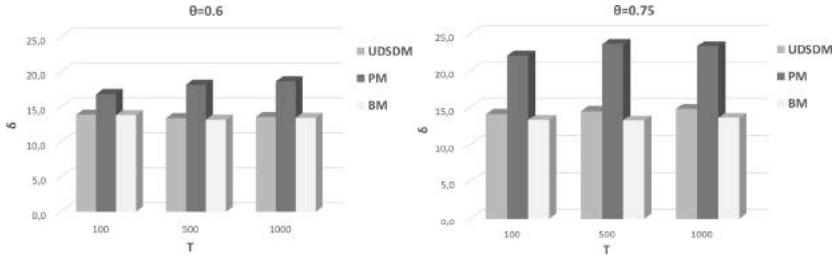


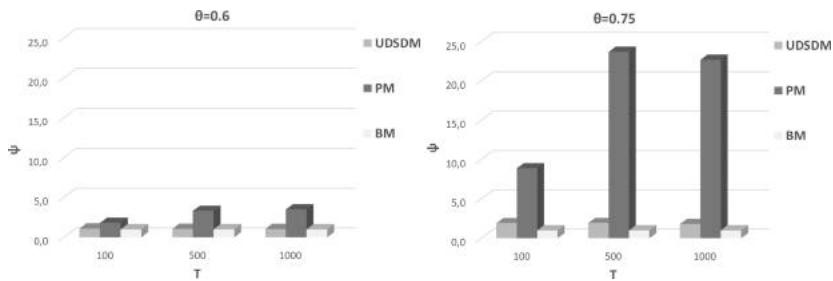
Fig. 2. Comparative results for the ϕ metric.

Figure 3 presents our results related to the δ metric, i.e., the update quanta at the time when the dissemination action is decided. We observe that the UDSDM requires a lower magnitude than the PM and higher or equal than the BM before it concludes the dissemination action. When $\theta = 0.6$, there is ‘stability’ of the required δ before the dissemination action. When $\theta = 0.75$, δ increases together with T . The PM requires a higher synapses magnitude to be collected compared to the remaining two models. These result present the ‘attitude’ of the proposed model to wait and aggregate update quanta in order to alleviate the network from an increased number of messages. However, our model does not wait till the expiration of T to report fresh synapses to the network. We can easily observe that the proposed model relies in the middle between the BM and the PM (with an ‘attitude’ to be close to the BM).

In Fig. 4, we present our results related to the ψ metric. We confirm our observations obtained by the two above discussed metrics, i.e., the UDSDM relies

**Fig. 3.** Comparative results for the δ metric.

in the middle between the BM and the PM. Our model is mainly affected by the rationale to distribute fresh synapses in the burden of the number of messages circulated in the network. However, it manages to deliver less messages than the BM. We observe a stability in the obtained outcomes exhibiting the capability of the UDSDM to detect changes in synapses quanta and fire the delivery action. The PM requires the less frequency of the delivery, however, in the burden of the freshness of the distributed synapses.

**Fig. 4.** Comparative results for the ψ metric.

6 Conclusions

Data management at the edge of the network is a significant research subject due to the reduced latency that end users can enjoy if processing is performed at the EC infrastructure. Numerous IoT devices report data towards the Cloud data-center, thus, advanced data management applications should be provided at the EC as the intermediate point where processing can take place. A number of EC nodes may undertake the responsibility of hosting datasets and processing activities. Nodes should act in a collaborative manner to increase their performance. For instance, EC nodes may exchange processing tasks or data to conclude the desired outcomes as soon as possible. In this paper, we enhance the collaborative

aspect of the EC infrastructure and propose a novel model for exchanging data synopses at the edge of network. The target is to have all EC nodes informed about the data present at their peers, thus, to take optimal decisions related to the management of the requested processing activities. We present a deep learning model and an uncertainty driven scheme to reason over the appropriate time to exchange data synopses. The deep learning model manages to learn the distribution of the concluded synopsis as the basis for retrieving future estimations. The uncertainty driven scheme deals with a set of rules applied upon past synopses observations and future estimates. This way, we combine two completely different technologies to realize an efficient system for the management of data synopses at EC. Our aim is to provide a decision making methodology that minimizes the number of messages circulated in the network, however, without jeopardizing the freshness of the exchanged statistical information. We discuss our model adopting the principles of the FL and present the relevant formulations. EC nodes monitor their data and decide when it is the right time to deliver the current data synopsis. Our experimental evaluation shows that the proposed scheme can efficiently assist in the envisioned goals being evidenced by numerical results. In the first place of our future research plans, it is to incorporate a rewarding mechanism for every ‘correct’ decision and present a system that learns on how to learn. Additionally, we want to involve more parameters in the decision making mechanism like a ‘snapshot’ of the current status of every EC node.

References

1. Aggarwal, C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. In: VLDB Conference, pp. 81–92 (2003)
2. Aggarwal, C., Han, J., Wang, J., Yu, P.: On-demand classification of data streams. In: ACM KDD Conference, pp. 503–508 (2004)
3. Aggarwal, C., Yu, P.: A survey of synopsis construction in data streams. In: Aggarwal, C. (ed.) *Data Streams, Models and Algorithms*. Springer, Heidelberg (2007)
4. Alon, N., Gibbons, P., Matias, Y., Szegedy, M.: Tracking joins and self joins in limited storage. In: ACM PODS Conference, pp. 10–20 (1999)
5. Amrutha, S., et al.: Data dissemination framework for IoT based applications. Indian J. Sci. Technol. **9**(48), 1–5 (2016)
6. Anagnostopoulos, C., Kolomvatsos, K.: An intelligent, time-optimized monitoring scheme for edge nodes. J. Netw. Comput. Appl. **148** (2019). <https://doi.org/10.1016/j.jnca.2019.102458>
7. Anglano, C., Canonico, M., Guazzone, M.: Profit-aware resource management for edge computing systems. In: 1st International Workshop on Edge Systems, Analytics and Networking, pp. 25–30 (2018)
8. Babcock, B., Datar, M., Motwani, R.: Load shedding techniques for data stream systems. In: Workshop on Management and Processing of Data Streams (2003)
9. Babcock, B., Babu, S., Datar, M., Motwani, R., Widom, J.: Models and issues in data stream systems. In: PODS, pp. 1–16 (2002)
10. Bellavista, P., Corradi, A., Foschini, L., Scotese, D.: Differentiated service/data migration for edge services leveraging container characteristics. IEEE Access **7** (2019)

11. Bhardwaj, K., Agrawal, P., Gavrilovska, A., Schwan, K.: AppSachet: distributed app delivery from the edge cloud. In: 7th International Conference Mobile Computing, Applications, and Services, pp. 89–106 (2015)
12. Chakrabarti, K., Garofalakis, M., Rastogi, R., Shim, K.: Approximate query processing with wavelets. VLDB J. **10**(2–3), 199–223 (2001)
13. Charikar, M., Chen, K., Farach-Colton, M.: Finding frequent items in data streams. In: ICALP (2002)
14. Cherrueau, R.A., Lebre, A., Pertin, D., Wuhib, F., Soares, J.: Edge computing resource management system: a critical building block!. In: USENIX Workshop on Hot Topics in Edge Computing, Initiating the debate via OpenStack, pp. 1–6 (2018)
15. Chu, D., Deshpande, A., Hellerstein, J., Hong, W.: Approximate data collection in sensor networks using probabilistic models. In: 22nd International Conference on Data Engineering (ICDE 06) (2006)
16. Cormode, G., Muthukrishnan, S.: What's hot and what's not: tracking most frequent items dynamically. In: ACM PODS Conference (2005). <https://doi.org/10.1145/1061318.1061325>
17. Dobra, A., Garofalakis, M.N., Gehrke, J., Rastogi, R.: Sketch-based multi-query processing over data streams. In: EDBT Conference (2004). https://doi.org/10.1007/978-3-540-24741-8_32
18. Gehrke, J., Korn, F., Srivastava, D.: On computing correlated aggregates over continual data streams. In: SIGMOD Conference (2001). <https://doi.org/10.1145/375663.375665>
19. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
20. Hagras, H.: A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots. IEEE TFS **12** (2004). <https://doi.org/10.1109/TFUZZ.2004.832538>
21. Karanika, A., Oikonomou, P., Kolomvatsos, K., Loukopoulos, T.: A demand-driven, proactive tasks management model at the edge. In: IEEE International Conference on Fuzzy Systems (FUZZ-IEEE) (2020)
22. Kolomvatsos, K.: A proactive uncertainty driven model for data synopses management in pervasive applications. In: 6th IEEE International Conference on Data Science and Systems (DSS), Fiji, 14–16 December (2020)
23. Kolomvatsos, K.: An intelligent scheme for assigning queries. Appl. Intell. **48**(9), 2730–2745 (2017). <https://doi.org/10.1007/s10489-017-1099-5>
24. Kolomvatsos, K.: A distributed, proactive intelligent scheme for securing quality in large scale data processing. Computing **101**, 1687–1710 (2019)
25. Kolomvatsos, K., Anagnostopoulos, C.: An intelligent edge-centric queries allocation scheme based on ensemble models. ACM Trans. Internet Technol. (2020). <https://doi.org/10.1145/3417297>
26. Kolomvatsos, K., Anagnostopoulos, C.: A probabilistic model for assigning queries at the edge. Computing **102**, 865–892 (2020)
27. Kolomvatsos, K., Anagnostopoulos, C.: Multi-criteria optimal task allocation at the edge. Futur. Gener. Comput. Syst. **93**, 358–372 (2019)
28. Kolomvatsos, K., Anagnostopoulos, C., Hadjiefthymiades, S.: Data fusion & type-2 fuzzy inference in contextual data stream monitoring. IEEE Trans. Syst. Man Cybern.: Syst. **PP**(99), 1–15 (2016)
29. Kolomvatsos, K., Anagnostopoulos, C., Koziri, M., Loukopoulos, T.: Proactive & Time-Optimized Data Synopsis Management at the Edge. IEEE Trans. Knowl. Data Eng. (IEEE TKDE) (2020). <https://doi.org/10.1109/TKDE.2020.3021377>

30. Kolomvatsos, K., Anagnostopoulos, C., Marnerides, A., Ni, Q., Hadjiefthymiades, S., Pezaros, D.: Uncertainty-driven ensemble forecasting of QoS in software defined networks. In: 22nd IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece (2017)
31. Lakshmi, K.P., Reddy, C.R.K.: A survey on different trends in data streams. In: IEEE International Conference on Networking and Information Technology (2010). <https://doi.org/10.1109/ICNIT.2010.5508473>
32. Manku, G., Motwani, R.: Approximate frequency counts over data streams. In: VLDB Conference (2002)
33. Martin, R., Vahdat, A., Culler, D., Anderson, T.: Effects of communication latency, overhead, and bandwidth in a cluster architecture. In: 4th Annual International Symposium on Computer Architecture (1997). <https://doi.org/10.1145/384286.264146>
34. Mendel, J.M.: Type-2 fuzzy sets and systems: an overview. IEEE Comput. Intell. Mag. **2**(1) (2007). <https://doi.org/10.1109/MCI.2007.380672>
35. Mendel, J.M.: Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions. Prentice-Hall, Upper Saddle River (2001)
36. Mesiar, R., Kolesarova, A., Calvo, T., Komornikova, M.: A review of aggregation functions. Studies in Fuzziness and Soft Computing (2008). https://doi.org/10.1007/978-3-540-73723-0_7
37. Muthukrishnan, S.: Data streams: algorithms and applications. In: 14th Annual ACM-SIAM Symposium on Discrete Algorithms (2003)
38. Najam, S., Gilani, S., Ahmed, E., Yaqoob, I., Imran, M.: The role of edge computing in Internet of Things. IEEE Commun. Mag. (2018). <https://doi.org/10.1109/MCOM.2018.1700906>
39. Novák, V., Perfilieva, I., Močkoř, J.: Mathematical Principles of Fuzzy Logic. Kluwer Academic, Dordrecht (1999)
40. Sardellitti, S., Scutari, G., Barbarossa, S.: Joint optimisation of radio and computational resources for multicell mobile-edge computing. IEEE Trans. Signal Inf. Process. Netw. **1**(2), 89–103 (2015)
41. Savolainen, P., et al.: Spaceify: a client-edge-server ecosystem for mobile computing in smart spaces. In: International Conference on Mobile Computing & Networking, pp. 211–214 (2013)
42. Schweller, R., Gupta, A., Parsons, E., Chen, Y.: Reversible sketches for efficient and accurate change detection over network data streams. In: Internet Measurement Conference Proceedings, pp. 207–212 (2004)
43. Shekhar, S., Gokhale, A.: Dynamic resource management across cloud-edge resources for performance-sensitive applications. In: 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (2017)
44. Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Satyanarayanan, M.: Scalable crowd-sourcing of video from mobile devices. In: 11th Annual International Conference on Mobile Systems, Applications, and Services, pp. 139–152 (2013)
45. Tatbul, N., Zdonik, S.: A subset-based load shedding approach for aggregation queries over data streams. In: 32nd International Conference on Very Large Data Bases, Seoul, Korea (2006)
46. Vandeput, N.: Data Science for Supply Chain Forecast (2018). Independently Published
47. Wang, N., Varghese, B., Matthaieu, M., Nikolopoulos, D.: ENORM: a framework for edge node resource management. IEEE Trans. Serv. Comput. (2017). <https://doi.org/10.1109/TSC.2017.2753775>

48. Yao, Y., Cao, Q., Vasilakos, A.V.: EDAL: an energy-efficient, delay-aware, and lifetime-balancing data collection protocol for wireless sensor networks. In: IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, pp. 182–190 (2013)
49. Zhou, A., Wang, S., Li, J., Sun, Q., Yang, F.: Optimal mobile device selection for mobile cloud service providing. *J. Supercomput.* **72**(8), 3222–3235 (2016)



DeepObfusCode: Source Code Obfuscation through Sequence-to-Sequence Networks

Siddhartha Datta^(✉)

University of Oxford, Oxford, UK
siddhartha.datta@cs.ox.ac.uk

Abstract. The paper explores a novel methodology in source code obfuscation through the application of text-based recurrent neural network (RNN) encoder-decoder models in ciphertext generation and key generation. Sequence-to-sequence models are incorporated into the model architecture to generate obfuscated code, generate the deobfuscation key, and live execution. Quantitative benchmark comparison to existing obfuscation methods indicate significant improvement in stealth and execution cost for the proposed solution, and experiments regarding the model's properties yield positive results regarding its character variation, dissimilarity to the original codebase, and consistent length of obfuscated code.

Keywords: Code obfuscation · Encoder-decoder models

1 Introduction

The field of code obfuscation has aimed to tackle reverse-engineering of code bases for years. The entire basis of this methodology is that if a program is constructed with logic not easily recognizable by a reader, the logic would be preserved intact and the software would be intrinsically protected. Traditional tactics include creative uses of whitespace, redundant logical operations, unnecessary conditional operations, amongst others. The common issue with obfuscation is that it can be reverse-engineered, the only factor for a malicious actor would be the amount of time needed to discover the logic. *DeepObfusCode* is a proposed methodology to use neural networks to convert the plaintext source code into a cipher text by using the propagating architecture of neural networks to compound the randomness factor in the creation of the ciphertext. Yet at the same time, neural networks have the ability to learn statistical patterns and generate weights to convert one text to another, in our case from the ciphertext to plaintext. This would eventually permit users to simply load the ciphertext and the key to self-execute the program without foreign users viewing

S. Datta—Work performed at the Hong Kong University of Science and Technology.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2021
K. Arai (Ed.): *Intelligent Computing*, LNNS 284, pp. 637–647, 2021.

https://doi.org/10.1007/978-3-030-80126-7_45

the inner workings of the code. From an academic standpoint, this methodology redirects obfuscation methodology towards complete obfuscation in contrary of incremental obfuscation, and suggests the usage and development of asymmetric key infrastructure in obfuscation. Beyond sequence-to-sequence network models, further obfuscation models could be built with greater degree of resilience, and other deep learning methods could be harnessed to develop alternative techniques to obfuscate code. The methodology can be adopted for more efficient, more effective obfuscation for developers protecting their proprietary codebase or cloud computing services wishing to guarantee confidentiality to customers. The algorithmic architecture could be further incorporated into larger frameworks or infrastructures to render homomorphic encryption and ensure complete anonymity of the codebase during execution from the execution provider.

2 Related Work

This section will detail existing code obfuscation methods being used, how they are evaluated, and how modern deep learning techniques are being used in the area, while drawing comparison to the proposed obfuscation implementation.

2.1 Code Obfuscation Methods

The objective of code obfuscation is to mask the computational processes and logic behind software code, to protect trade secrets, intellectual property, or confidential data. As such, there have been eight generalized methods behind code obfuscation [1], namely, (i) Name obfuscation; (ii) Data obfuscation; (iii) Code flow obfuscation; (iv) Incremental obfuscation; (v) Intermediate code optimization; (vi) Debug information obfuscation; (vi) Watermarking; (vii) Source code obfuscation. Source code obfuscation, the focus of the paper, is the process that hides the meaning behind the source code, in the case that a third party obtains the code that the code renders itself un-understandable. Under each branch of obfuscation method, there are sub-techniques to reduce understandability of code that are shared by the other obfuscation methods, including control ordering (changing the execution order of program statements), control computation (changing the control flow of the program, such as inserting dead code to increase code complexity), data aggregation (changing the access to data structures after their conversion into other types), renamed identifiers (replacing identifiers with meaningless strings as new identifiers to reduce readability and comprehension of what certain functions or methods do). While existing methods tend to require manual altering of source code with the aforementioned methods, the proposed method performs complete obfuscation with relatively randomly generated characters as the code output. Malicious attackers or readers of the code will not be able to reverse-engineer the code based on the readability of the obfuscated code, and a well-lodged key file of the model weights (e.g. kept on the execution server) would prohibit de-obfuscation.

Existing methods of evaluating obfuscation techniques have tended towards qualitative surveys of difficulty and time taken to de-obfuscate by students,

software engineers or computer scientists [2]. To quantitatively compare the performance of the proposed source code obfuscation method against existing code obfuscation methods, we will modify a framework that has been used to compare obfuscation methods before [3], but also malleable enough for us to adapt for our specific comparison use case. The four original comparison metrics are:

- (i) **Code Potency:** This metric arbitrarily computes the degree of obfuscation with traditional complexity measures, with specific focus on control flow and data obfuscation. An example of an implementation would be frequency counts of misleading statements.
- (ii) **Resilience:** This metric measures the ability of the obfuscated text to withstand attacks from automated tools.
- (iii) **Stealth:** This metric tests the difficulty in manual de-obfuscation by humans. It inherently checks how quickly adversarial de-obfuscators can detect misdirecting statements, correctly interpret supposedly-confusing identifiers, and tests their ability to uncover the logic and process behind the code without prior or minimal knowledge of the program.
- (iv) **Execution Cost:** This metric measures (i) the incremental time required to perform the obfuscation, and (ii) the incremental time required to execute the obfuscated code. This would be contrasted to the time taken for the original code without obfuscation.

Since the proposed obfuscation method generates a ciphertext completely different from the original source text and cannot be reverse-engineered by manual human de-obfuscation, the former two metrics (code potency and resilience) would not be used as comparison metrics for this method. The main metrics for evaluating *DeepObfusCode* would be stealth and execution time.

2.2 Applications of Neural Networks

Neural networks have had recent applications in the field of cryptography and stenography. There has been a recent implementation of style transfer in stenography [4], where secret images are encrypted within a cover image, and the secret could be obtained by passing the encrypted image through a secondary network. There is another implementation of n-dimensional convolutional neural networks used to encrypt input data [5]. Another implementation involves the use of a multilayer perception (MLP) model to encrypt satellite images, and decrypting using a secondary MLP model [6]. Implementations on neural networks that can execute on encrypted data has also been a recent development [7]. This indicates a growing interest in encrypting data through the use of neural networks. Unlike prior on data encryption through deep learning techniques, the proposed architecture opens an alternative application, which is to encrypt source code itself through the use of deep learning, then using the generated model file to decrypt and execute the code.

RNN Encoder-Decoders or sequence-to-sequence models, trained to predict an output text set from provided input text, have had growing applications in statistical machine translation [8], grammatical error correction [9],

and text summarization [10]. The notion of taking input sentences and converting them to labeled output sentences with a trained model of weights for character-by-character prediction would be the basis for the proposed obfuscation method. The proposed obfuscation utilizes the advantage of text-based encoder-decoders in character generation and calculating weights to convert one string into another.

3 DeepObfusCode

The architecture behind the obfuscation is first a primary recurrent neural network (RNN) encoder-decoder model with randomly-set weights taking the original code text as an input to generate the obfuscated text. Then a secondary RNN encoder-decoder model is passed two arguments, the generated ciphertext and the original code as the corresponding label, and is trained for a number of iterations to calculate weights to generate the original text, and the weights of the network would be the key generated. This section will discuss the architecture and implementation in further detail. Supporting code is available.¹

3.1 Ciphertext Generation

To generate the obfuscated code, we first take the original legible text (source code) and a full character set (a string containing all characters, including letters, numbers and punctuation) (Fig. 1).

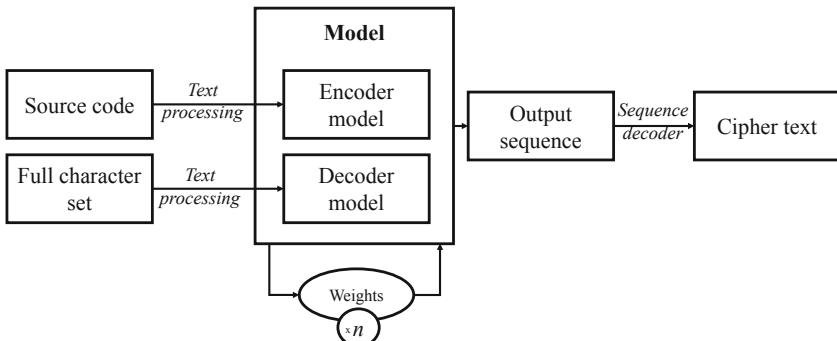


Fig. 1. Overview of ciphertext generation. We pass the source code and a character set as inputs to initialize both character sets for the encoder-decoder model, then randomly assign weights to generate the ciphertext, an obfuscated version of the source code.

We perform text processing for both text strings to prepare the inputs for the encoder and decoder models. First we create unique character sets for both

¹ Code repository: <https://github.com/dattasiddhartha/DeepObfusCode>.

strings, then create two dictionaries for both strings to index each character (key is index and value is character), and create two other dictionaries for both strings to return character given its index (key is character and value is index). Finally we vectorize both strings.

We use the source code character set as inputs for the encoder model and both text character sets for the decoder model. We will randomly generate weights for a given number of times n (the randomness index; in our experiment we set $n = 10$), and set the model weights to be the generated weights array. The variation in the weights array will determine the degree of randomness in the ciphertext.

Weights generated will alter the character value generated for each segment of the string, as shown in the Ciphertext Generation function $C(p)$. c refers to the ciphertext (obfuscated code), p refers to the plaintext (source code), N refers to each weight position. f_n and w_{rand} are the n -th feature and randomized weight, respectively. $Z(p)$ is a normalization constant that is independent of the weights.

$$C(p) = \log p(c|p) = \sum_{n=1}^N w_{rand} f_n(c|p) + \log Z(p)$$

The output will be the output sequence, to which we will decode by taking reference from the index-to-character dictionary to convert the output sequence into an output character string and iteratively append characters. The resulting output would be the ciphertext.

3.2 Key Generation

After generating the ciphertext, we use it along with the original source code plaintext to generate a key (Fig. 2). We apply the same text processing steps as before, set ciphertext as input into the encoder, and both texts as inputs into the decoder, and train the model for a certain number of iteration counts. For our experiments, we tended to use 2000 iterations, though Early Stopping mechanisms could be used to lower training time for shorter source code; the iteration count should be the number of iterations needed to ensure the output text is executable and identical to the source code text. After the training and model weight setting process is complete, export the encoder and decoder model files in HDF5 format (the key), and export the metadata (the dictionaries of index to char, and char to index) in pickle format. The Key Generation function $K(p, c)$ accepts the arguments c ciphertext and p plaintext, and sets weights while minimizing the loss function. f_n and w_n are the n -th feature and weight (after training), respectively. $Z(c)$ is a normalization constant that is independent of the weights.

$$\max \frac{1}{N} \sum_{n=1}^N \log p(p|c)$$

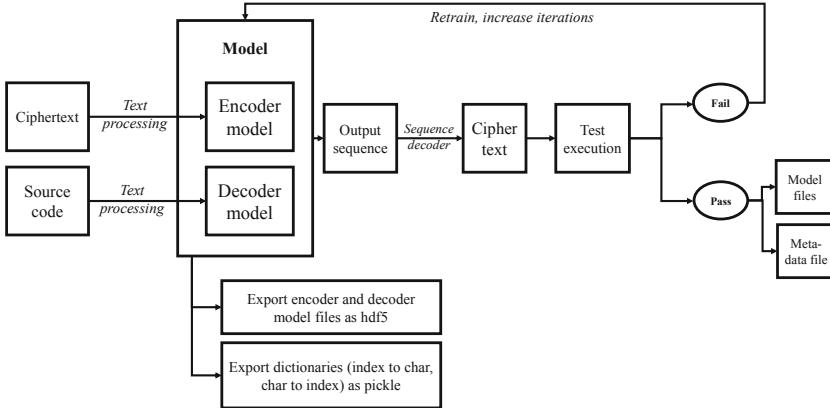


Fig. 2. Overview of key generation. With the known ciphertext and original source code, the developer of the source code would pass them as inputs into another encoder-decoder model and train over a number of iterations such that the model weights obtained can translate the obfuscated code into executable code, with validation of executability at the end.

$$K(p, c) = \log p(p|c) = \sum_{n=1}^N w_n f_n(p|c) + \log Z(c)$$

After the ciphertext is decoded from the output sequence, we test if the code is executable; if it fails, we retrain the model (a pre-determined loss threshold that ensures correct execution would facilitate early stopping in iterative training); if it passes, the ciphertext and key pair are retained.

3.3 Source Code Execution

During live execution (Fig. 3), we have three inputs: the obfuscated code (ciphertext), the key (model files), and the metadata files; function $K(c, k)$ would be modified to accept c ciphertext, and k model file (along with metadata file). For our own experiments, the model and metadata files were separate; for execution in live systems, it would be possible to combine them into a single file if preferred. When we pass all three through the model container, the output value is executed as soon as it is returned, i.e. $Exec(K(c, k))$.

4 Results

The evaluate the obfuscation method compared to existing obfuscation techniques, the method will be tested on two aspects, stealth and execution cost. Other parameters such as code potency or resilience would not be applicable to this method, as those comparison metrics require some form of the original code to be preserved; but since this obfuscation method completely regenerates

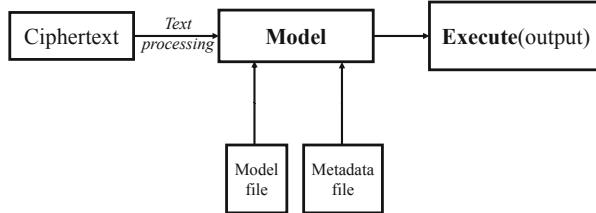


Fig. 3. Overview of live execution. To run the obfuscated code on any server or system, one would pass in the obfuscated code into an execution engine that takes the ciphertext and the lodged model files as inputs to execute the withheld code.

the code base to be indistinguishable from the original code, such testing (e.g. searching for misleading control flow) will not work.

4.1 Stealth

The objective of testing for the stealthiness of obfuscated code is to measure the complexity or distance in randomness from the original code. Hence the logic behind our comparison is to first obtain recognized well-obfuscated code and its deobfuscated copy (the original code), obfuscate the original code using our proposed method, then compare the distance in similarity between the original code and each of the obfuscated code.

Obfuscated code samples were obtained through the International Obfuscated C Code Contest (IOCCC), and de-obfuscated code samples of the selected obfuscated code was found on forums. The obfuscated samples were winners in the IOCCC contest, signifying high quality, and were manually created by programmers. A total of five obfuscated-deobfuscated paired samples were used for the experiment.² Obfuscated code of reputable, unbiased quality and its deobfuscated counterpart is generally difficult to obtain in large samples; the aforementioned samples would be used in bench-marking and comparison, but not for testing the inherent properties of the proposed architecture.

To test the obfuscation method extensively, for each de-obfuscated sample, we performed the obfuscation to obtain the ciphertext for 100 iterations (to return 500 samples of obfuscated code samples in total, 100 per de-obfuscated code sample).

Then we used Levenshtein Distance to measure the distance between two code samples; the distance is the number of deletions, insertions, or substitutions required to transform the source string into the target string. After calculating the Levenshtein Distance values between the original de-obfuscated code and the IOCCC obfuscated code, we calculated the Levenshtein Distance values between the original de-obfuscated code with each respective sample's *DeepObfusCode* obfuscated code and took the mean Levenshtein Distance for the sample. The results are tabulated in Table 1.

² Dataset: <https://github.com/dattasiddhartha-1/obfuscation-dataset>.

Table 1. Levenstein distance comparison

Deobfuscated code set	Benchmark obfuscation lev. distance	Proposed obfuscation lev. distance	Ratio of proposed: benchmark
1	3061	4000.68687	1.30699
2	2587	497.17172	0.19218
3	42	102.81818	2.44805
4	4649	5780.32323	1.24335
5	9132	10195.40404	1.11645

From the table, we can observe that in general the *DeepObfusCode* obfuscation has a greater degree of randomness or dissimilarity from the original source text compared to the benchmark IOCCC obfuscated text, with an average magnitude improvement of 1.2614 (the average of the ratio of proposed:benchmark across all samples) and with a standard deviation 0.7176.

Set 2 of the experiment indicates the benchmark outperforming the proposed method. Inspection of set 2 reveals a greater proportion of repetitive characters as junk code insertions, which serve to confuse de-obfuscators, but also dilute the similarity (conversely inflate the Levenshtein distance) since the proportion of overlapping characters to the total number of characters would be lower. While the total collection indicates a general improvement in dissimilarity, removing set 2 would indicate an average magnitude improvement of 1.5287 and a standard deviation of 0.5352. This aids in justifying that the proposed obfuscation method performs well in the aspect of stealth.

4.2 Execution Cost

The primary focus of the section would be to measure how ciphertext generation time (time to encrypt) and key generation time (time to decrypt) would vary with the length of the source code (plaintext length) (see Fig. 4 and 5).

The experimental design starts with the random generation of strings varying from length 1 to 4000 (to vary the plaintext length). Then we set loggers to record information regarding the six main variables we would like to test: (i) Length of string input, (ii) Randomness metric (Levenshtein distance), (iii) Execution time for encryption, (iv) Execution time for decryption, (v) Character variation, (vi) Average character length of ciphertext. As execution time depends on the device running the simulation, for reference, the simulation was run on a Python Jupyter notebook, running on Windows 10 with a Nvidia Geforce 1070 GTX graphics card (running at 10% GPU usage level).

The plot of ciphertext generation time requirement against length of the source code shows no distinct pattern, while the plot of key generation time requirement against length of the source code shows a linear pattern. This infers that ciphertext generation is not length-dependent and can be executed with-

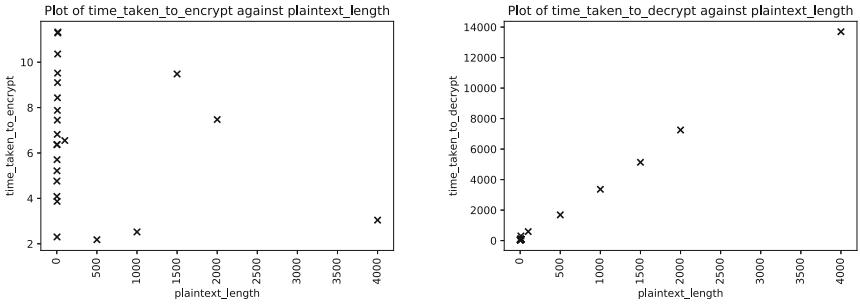


Fig. 4. Plots of time properties against plaintext length

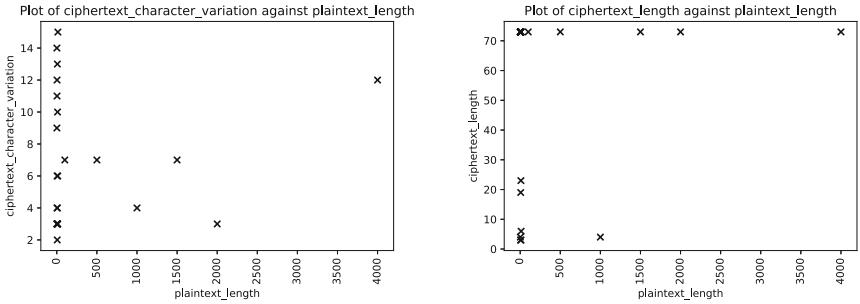


Fig. 5. Plots of ciphertext properties against plaintext length

out significant incremental cost. Since the length of the source code affects the training time required for the same number of iterations (higher length would increase training time per epoch), longer source code would require more time to generate a key, so this obfuscation method may be more suitable for smaller code bases or systems with sufficient computing resources. We can also infer the key generation takes linear time, i.e. time complexity is $O(n)$.

Beyond execution cost, this experiment yielded additional information regarding the properties of this obfuscation model.

Figure 6 adds onto prior Stealth results in Sect. 4.1 to reveal that the larger the code base, the greater the dissimilarity between the obfuscated code and the original code base.

Plotting ciphertext character variation and ciphertext length against plaintext length reveals: (i) the character variation is widely distributed regardless of the length of the plaintext input, which further supports the notion of randomness of ciphertext generation, as the ciphertext is based purely on the randomness in the model weight generation; (ii) the ciphertext length is kept low (on average 72 character length) regardless of the plaintext length, which reduces obfuscated code storage requirements. The notion that the cipher generation algorithm produces short random ciphertexts further implies it would be difficult for malicious actors to reverse-engineer the ciphertext by setting random weights or training a

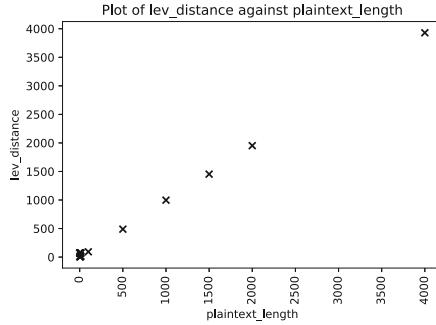


Fig. 6. Plot of similarity metric against plaintext length

model without a known output text. For the model used in the experiment, the model file consists of 3 layers of arrays: the main layer contains 8 arrays, of which each array has an array-length as represented in [39, 256, 1024, 72, 256, 1024, 256, 72], of which each sub-array (except the third and sixth array) contains an array of 1024 values. One would have to randomly generate 32-bit floats for 975,872 values, at least to a proximate range to the actual values before they can generate readable de-obfuscated code, but even then they would need to further generate values to the 8th decimal place if they intend to obtain scrambled text without any lapse in meaning.

The correlation matrix in Fig. 7 summarizes the relationships between the properties tested in the experiment.

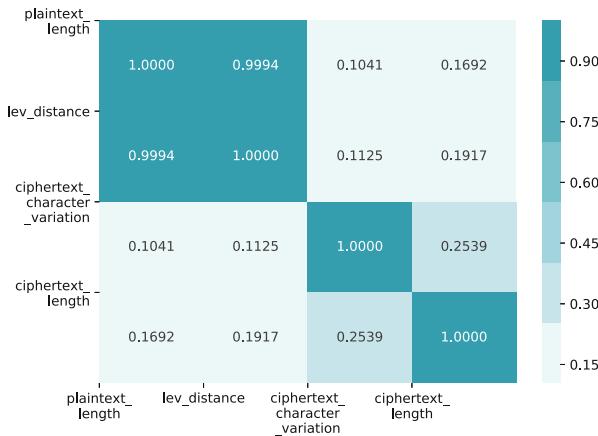


Fig. 7. Correlation matrix of properties

5 Conclusion

This paper presents a novel obfuscation methodology using RNN encoder-decoder models to generate ciphertext from source code and generating and utilizing model weights as keys. Compared to current obfuscation methods, it is at least on par in terms of stealth and is expected to outperform for larger code bases in terms of obscurity and readability, and though key generation may take a significant amount of time for larger code bases or require more computational resources, it would be less time-intensive than to manually obfuscate the source code. This would be a good use case application for services that have confidential source code in plaintext but would prefer ciphertext yet require the ability to execute.

References

1. Popa, M.: Techniques of program code obfuscation for secure software. *J. Mob. Embed. Distrib. Syst.* **3**, 205–219 (2011)
2. Viticchie, A., et al.: Assessment of Source Code Obfuscation Techniques (2017). <https://arxiv.org/pdf/1704.02307.pdf>
3. Schneider, J., Locher, T.: Obfuscation using Encryption (2016). <https://arxiv.org/pdf/1612.03345.pdf>
4. Baluja, S.: Hiding images in plain sight: deep steganography. In: Advances in Neural Information Processing Systems (2017)
5. Benoit, S.: ConvCrypt (2018). <https://github.com/santient/convcrypt>
6. Ismail, A., Galal-Edeen, H., Khattab, S., Mohamed, A.E., Bahtiy, M.E.: Satellite image encryption using neural networks backpropagation. In: International Conference on Computer Theory and Applications (2012)
7. Hesamifard, E., Takabi, H., Ghasemi, M.: CryptoDL: Deep Neural Networks over Encrypted Data (2017). <https://arxiv.org/pdf/1711.05189.pdf>
8. Cho, K., et al.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation (2014). <https://arxiv.org/pdf/1406.1078.pdf>
9. Ahmadi, S.: Attention-based Encoder-Decoder Networks for Spelling and Grammatical Error Correction (2018). <https://arxiv.org/pdf/1810.00660.pdf>
10. Khatri, C., Singh, G., Parikh, N.: Abstractive and extractive text summarization using document context vector and recurrent neural networks. In: KDD Deep Learning Day (2018)



Deep-Reinforcement-Learning-Based Scheduling with Contiguous Resource Allocation for Next-Generation Wireless Systems

Shu Sun^(✉) and Xiaofeng Li

Intel Corporation, Santa Clara, CA 95054, USA
{shu.sun,xiaofeng.li}@intel.com

Abstract. Scheduling plays a pivotal role in multi-user wireless communications, since the quality of service of various users largely depends upon the allocated radio resources. In this paper, we propose a novel scheduling algorithm with contiguous frequency-domain resource allocation (FDRA) based on deep reinforcement learning (DRL) that jointly selects users and allocates resource blocks (RBs). The scheduling problem is modeled as a Markov decision process, and a DRL agent determines which user and how many consecutive RBs for that user should be scheduled at each RB allocation step. The state space, action space, and reward function are delicately designed to train the DRL network. More specifically, the original quasi-continuous action space, which is inherent to contiguous FDRA, is refined into a finite and discrete action space to obtain a trade-off between the inference latency and system performance. Simulation results show that the proposed DRL-based scheduling algorithm outperforms other representative baseline schemes while having lower online computational complexity.

Keywords: Deep Reinforcement Learning (DRL) · Frequency-Domain Resource Allocation (FDRA) · Next generation · Scheduling

1 Introduction

Resource allocation is an indispensable ingredient in wireless systems with multiple user equipments (UEs), in order to meet certain quality of service (QoS) requirements such as throughput, fairness, latency, and/or reliability. The 3rd Generation Partnership Project (3GPP) has specified two types of downlink frequency-domain resource allocation (FDRA), i.e., type 0 and type 1, for the fifth-generation (5G) and beyond-5G (B5G) wireless communications [5, 6]. There exist two essential discrepancies between type-0 and type-1 FDRA: (1) Type 0 is on the resource block group (RBG) level, where an RBG contains a number of consecutive resource blocks (RBs) and an RB is defined as 12 consecutive subcarriers in the frequency domain [3], while type 1 is on the RB level;

- (2) The resources (RBGs or RBs) assigned to each UE can be non-contiguous for type 0, while they must be contiguous for type 1.

The contiguous-RB constraint in type-1 FDRA renders it extremely difficult to find an optimal UE and resource allocation strategy except using brute-force search whose computational complexity is prohibitively high. Consequently, it is vital to propose sub-optimal scheduling algorithms with contiguous FDRA that have reasonable complexity hence implementable in practice. A myriad of contiguous FDRA scheduling approaches have been proposed previously (for instance, [12, 15, 16] and references therein). In particular, three practical scheduling algorithms with contiguous FDRA have been presented in [12], wherein one of the algorithms, joint allocation with dual ends (JADE), yields the best performance and outperforms an existing contiguous FDRA method representative in the industry [12, 16]. Nevertheless, albeit its relatively low complexity compared with prior schemes, JADE still incurs considerable complexity when UEs abound since its complexity scales with the square of the number of UEs. On the other hand, artificial intelligence (AI) has found a variety of applications in the field of wireless communications [8–10, 18–20, 22], which can help solve problems difficult to handle using conventional non-AI methods, improve efficiency and/or performance of existing solutions, or reduce instantaneous computation time while offering comparable performance, among other advantages. Therefore, it will be beneficial if AI techniques can be leveraged to schedule UEs and resources that yields performance comparable to or even better than JADE while inducing noticeably lower online computational complexity.

Given the fact that UE and resource scheduling can be modeled by a Markov decision process [10, 11], deep reinforcement learning (DRL), an important branch of machine learning belonging to AI, can be adopted to train an agent to offer optimal or superhuman performance while requiring significantly less instantaneous computational effort. DRL has been applied to solve resource allocation problems in a wide range of fields including Internet of Things [7], vehicular communications [21], heterogeneous cellular networks [23], and cloud radio access network [17], to name a few. None of the existing works, however, has tackled FDRA with the contiguity constraint which is a crucial resource allocation approach in the 3GPP specifications for 5G and B5G systems [3, 5, 6].

In this work, we propose a novel algorithm that jointly schedules UEs and contiguous frequency-domain resources based on DRL, which is named as STAR (Super Type-1 Allocation based on Reinforcement learning). Specifically, after trained offline, the DRL-based algorithm can make judicious decisions instantaneously on which UE and how many RBs for that UE shall be scheduled jointly at each allocation step. Major merits of the proposed algorithm are two-fold: (1) it can yield remarkable performance which is even superior to JADE, and (2) it enjoys considerably reduced online computational complexity as compared to JADE. To the authors' best knowledge, this is the first work that utilizes DRL to perform scheduling with contiguous FDRA.

The rest of this paper is organized as follows. The system model and problem formulation are presented in Sect. 2. Details on the proposed scheduling algo-

rithm capitalizing on DRL are provided in Sect. 3, including the state, action, and reward design, as well as complexity analysis and training process. Section 4 demonstrates the simulated performance of the proposed method. Conclusions and future work are drawn in Sect. 5.

2 System Model and Problem Formulation

We investigate a downlink cellular system comprising one next-generation nodeB (gNB) and K UEs indexed by the set $\mathcal{K} = \{0, \dots, K - 1\}$, where the UEs' traffic types can be diverse with distinct QoS requirements. The transmission bandwidth part (BWP) W is orthogonally divided into B RBs indexed by the set $\mathcal{B} = \{0, \dots, B - 1\}$. The payload for UE k is denoted by L_k . Two constraints exist in the aforementioned type-1 FDRA in 3GPP 5G and B5G specifications [5, 6]: (1) exclusivity, i.e., an RB can only be allocated to at most one UE in the same time resource; (2) contiguity, indicating that the RBs assigned to each UE must be contiguous.

For UE k on RB b , given the estimated channel $\mathbf{H}_{k,b}$ and precoding matrix codebook [5], the rank indicator (RI), precoding matrix indicator (PMI), and modulation and coding scheme (MCS) [5] can be obtained, e.g., using the scheme in [13], after which the transport block size (TBS) per slot, $TBS_{k,b}$, is computed based on the RI, PMI, and MCS. The achievable rate of UE k on RB b in each slot is $r_{k,b} = TBS_{k,b}$. Let \mathcal{B}_k denote the set of RBs allocated to UE k , the achievable rate of UE k over \mathcal{B}_k is $r_k = \sum_{b \in \mathcal{B}_k} r_{k,b}$. The scheduling metric (e.g., sum-rate, proportional fairness (PF) [14], among others) can be flexible depending upon the system requirement. In this paper, sum-rate is selected as the scheduling metric as an example. The optimization problem can be formulated as

$$(P1): \max_{\{\mathcal{B}_0, \dots, \mathcal{B}_{K-1}\} \subseteq \mathcal{D}} \sum_{k \in \mathcal{K}} \sum_{b \in \mathcal{B}_k} r_{k,b} \quad (1)$$

subject to $\mathcal{B}_k \leq \mathcal{B}_{k'} = *, \forall k \neq k', k, k' \in \mathcal{K},$

$d_k \leq \sigma_k, \forall k \in \mathcal{K}$

where \mathcal{D} is the set of all possible RB allocations satisfying the contiguity constraint, d_k and σ_k denote the head-of-line (HoL) delay and delay threshold (the maximum allowable delay from packet generation to packet scheduling) of UE k , respectively. Note that a packet will be dropped and will not contribute to the TBS if it is not entirely scheduled before its HoL delay exceeds its delay threshold. The optimal solution to (P1) requires exhaustive search with prohibitively high computational complexity. In [12], a sub-optimal algorithm JADE has been put forth to solve (P1), which has been proved to provide near-optimal performance based on simulation results. The procedures of JADE is detailed in Algorithm 1. The main design principle of JADE is to jointly prioritize the UE and RB(s) in each allocation step that produces the largest scheduling metric with the minimum number of RBs, where the RB selection is performed and compared between both ends of the active BWP to exploit frequency diversity.

More specifically, for each UE, JADE first calculates the number of RBs needed to transmit its payload from both ends of the BWP, and selects the end of the BWP that requires fewer RBs, and computes the associated scheduling metric (i.e. sum-TBS herein). Then the UE possessing the largest scheduling metric is selected and its final MCS is calculated over the selected RBs. The steps above are executed iteratively until there is no remaining UE or RB.

Algorithm 1. Joint Allocation with Dual Ends (JADE) [12]

Require: Initialize $\mathcal{K}^\alpha = \leq \mathcal{B}^\alpha = \leq$

- 1: **while** $\mathcal{K} = \leq$ and $\mathcal{B} = \leq$ **do**
- 2: **for** $\forall k \in \mathcal{K}$ **do**
- 3: Calculate the number of RBs needed, $n_{k,\text{start}}$, to transmit L_k starting from the first remaining RB in \mathcal{B} and going forward, until $r_{k,\text{start}} \geq L_k$ or $\mathcal{B} = \leq$. Denote the selected RB set as $\mathcal{B}_{k,\text{start}}$.
- 4: Calculate the number of RBs needed, $n_{k,\text{end}}$, to transmit L_k starting from the last remaining RB in \mathcal{B} and going backward, until $r_{k,\text{end}} \geq L_k$ or $\mathcal{B} = \leq$. Denote the selected RB set as $\mathcal{B}_{k,\text{end}}$.
- 5: If $n_{k,\text{start}} \leq n_{k,\text{end}}$, store $\mathcal{B}_{k,\text{start}}$ and $r_{k,\text{start}}$ as \mathcal{B}_k and r_k , respectively; otherwise store $\mathcal{B}_{k,\text{end}}$ and $r_{k,\text{end}}$ as \mathcal{B}_k and r_k , respectively.
- 6: **end for**
- 7: $k^\alpha = \underset{k}{\operatorname{argmax}} r_k$.
- 8: Calculate MCS $_{k^*}$, the final MCS for UE k^α over \mathcal{B}_{k^*} .
- 9: $\mathcal{K}^\alpha \leftarrow \mathcal{K}^\alpha \cup \{k^\alpha\}$, $\mathcal{B}^\alpha \leftarrow \mathcal{B}^\alpha \cup \mathcal{B}_{k^*}$.
- 10: $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k^\alpha\}$, $\mathcal{B} \leftarrow \mathcal{B} \setminus \mathcal{B}_{k^*}$.
- 11: **end while**
- 12: **return** \mathcal{K}^α , \mathcal{B}^α , and MCS $_k$, $\forall k \in \mathcal{K}^\alpha$.

3 Proposed Scheduling Algorithm Based on DRL

It is noteworthy that in JADE, the number of scheduling metric calculation is proportional to K^2 due to the iteration for each remaining UE and RB. To reduce the online computational complexity and further enhance performance, we propose using a DRL-based scheduling approach, STAR, to smartly allocate RBs to multiple UEs. A deep Q-network (DQN) is employed for the training of STAR. The input and output relationship per slot incorporating DRL is illustrated in Fig. 1, where the green shaded modules represent the environment, and the blue module depicts the DRL agent. Each slot usually includes multiple allocation steps, wherein at each allocation step, a UE and a number of consecutive RBs are selected to be scheduled. Within each slot, a state is generated from the environment and is then passed to the DRL agent, after which the DRL agent takes an action and outputs to the environment, then a reward and the next state are produced from the environment based on the current state and action pair, and the steps above are repeated multiple times until there is no remaining

Algorithm 2. Super Type-1 Allocation based on Reinforcement Learning (STAR)

Require: Initialize $\mathcal{K}^\alpha = \leq \mathcal{B}^\alpha = \leq \mathbf{s}$ (to be detailed in Algorithm 3).

- 1: **while** $\mathcal{K} = \leq$ and $\mathcal{B} = \leq$ **do**
- 2: The DRL agent takes an action a based upon \mathbf{s} , where a indicates which UE and how many RBs should be allocated (see Table 1 for more detailed information). Denote the selected UE and selected RB set as k^α and \mathcal{B}_{k^*} , respectively.
- 3: The DRL agent passes a to the environment.
- 4: The environment generates a reward r and the next state $\tilde{\mathbf{s}}$ based on a , and passes them to the DRL agent.
- 5: Calculate MCS_{k^*} , the final MCS for UE k^α over \mathcal{B}_{k^*} .
- 6: $\mathbf{s} \leftarrow \tilde{\mathbf{s}}$.
- 7: $\mathcal{K}^\alpha \leftarrow \mathcal{K}^\alpha \cup \{k^\alpha\}$, $\mathcal{B}^\alpha \leftarrow \mathcal{B}^\alpha \cup \mathcal{B}_{k^*}$.
- 8: $\mathcal{K} \leftarrow \mathcal{K} \setminus \{k^\alpha\}$, $\mathcal{B} \leftarrow \mathcal{B} \setminus \mathcal{B}_{k^*}$.
- 9: **end while**
- 10: **return** \mathcal{K}^α , \mathcal{B}^α , and $\text{MCS}_k, \forall k \in \mathcal{K}^\alpha$.

UE or RB. The overall process of STAR is detailed in Algorithm 2. Theoretically, the RB allocation involves the determination of both the starting location and number of RBs, which is a highly entangled problem. To reduce the complexity of the DRL training and inference without significantly compromising the TBS performance, the starting location of RBs to be allocated is fixed as the first remaining RB in the BWP, so that each action only needs to determine which UE and how many RBs should be allocated without dealing with the RB starting location. The inclusion of the starting location of RB allocation is deferred to future work.

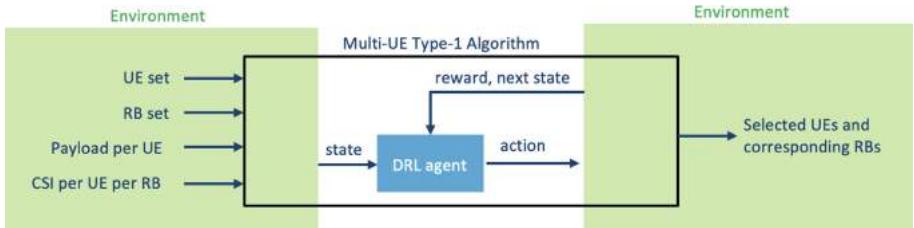


Fig. 1. Input and Output Relationship Per Slot for Multi-UE Type-1 FDRA based on DRL. The green Shaded Modules Represent the Environment, And the Blue Module Denotes the DRL agent.

3.1 State Design

In this work, as mentioned previously, DRL is utilized to determine which UE and how many RBs for the selected UE should be scheduled at each allocation step. The state design is closely related to the scheduling metric used in the

scheduling algorithm. Since the scheduling metric adopted herein is sum-rate, or equivalently, sum-TBS, both RB-related channel state information (CSI) and the metric related to sum-TBS (i.e., the remaining payload), should be reflected in the state. Detailed methodology for the acquisition of the current state \mathbf{s} at each RB allocation step is given in Algorithm 3. It is worth noting that if the next UE scheduling and resource allocation takes place in the current slot, the next state $\tilde{\mathbf{s}}$ is the state after the current UE scheduling and resource allocation in the current slot. In contrast, if the next UE scheduling and resource allocation occurs in the next slot, $\tilde{\mathbf{s}}$ takes the first state in the next slot, which reflects both the decision made in the current slot and new packet information and CSI in the next slot. This way, the state transition is always continuous and Markovian for both intra-slot and inter-slot cases.

Algorithm 3. State Design for STAR

```

1: for  $\forall k \in \mathcal{K}$  do
2:   for  $\forall b \in \mathcal{B}$  do
3:      $g_{k,b} = \text{MCS}_{k,b} \times \text{RI}_{k,b}$ 
4:   end for
5:   if  $L_k > 0$  then
6:      $\mathbf{s}_k = [L_k, g_{k,0}, g_{k,1}, \dots, g_{k,B-1}]$ 
7:   else
8:      $\mathbf{s}_k = \underbrace{[-1, -1, -1, \dots, -1]}_{B+1}$ 
9:   end if
10: end for
11:  $\mathbf{s} = [\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{K-1}]$ 
12: return  $\mathbf{s}$ 

```

3.2 Action Design

For contiguous FDRA, the action consists of two aspects: UE selection and RB allocation, which leads to a quasi-continuous action space \mathcal{A} due to potentially large numbers of RBs to be allocated. Considering the trade-off between training/inference overhead and performance, we transform the quasi-continuous action space into a discrete and finite action space. In particular, the size of the action space herein is devised to be $|\mathcal{A}| = 5K$, where the actions are indexed by $0, 1, \dots, 5K - 1$. The quotient produced by the division of the action index over 5 equals the UE index, i.e., $a//5 = k$, where a denotes the action index. For example, Actions 0 to 4 refer to selecting UE 0, Actions 5 and 9 indicate selecting UE 1, so on and so forth. In order to determine how many RBs should be allocated to each UE, the number of RBs needed to transmit the payload L_k , denoted as $n_{k,\text{WB}}$, is first calculated based on the wideband (WB) channel quality indicator (CQI) [5] of each UE, where the procedure in Sect. II of [12] is

Table 1. Action a and the Corresponding Physical Meaning

$a//5$	k
Selected UE	k
$a\%5$	0 1 2 3 4
n_k	$n_{k,\text{WB}} - 2$ $n_{k,\text{WB}} - 1$ $n_{k,\text{WB}}$ $n_{k,\text{WB}} + 1$ $n_{k,\text{WB}} + 2$

adopted to obtain the WB CQI. The number of RBs allocated to UE k , n_k , is given by

$$n_k = n_{k,\text{WB}} + a\%5 - 2 \quad (2)$$

where $a\%5$ represents the remaining resultant from the division of a over 5. Intuitively, the five actions associated with UE k are related to the number of RBs to be allocated based on $n_{k,\text{WB}}$. The overall actions and their meanings are listed in Table 1.

Note that variants of the action design described above can also be considered in practice according to specific overhead and/or performance requirements. For instance, the range of actions per UE can be smaller or larger than $5K$, i.e., n_k can be within ± 1 with respect to $n_{k,\text{WB}}$ to accelerate training and inference processes, or $\pm 3, \pm 4, \dots$ with respect to $n_{k,\text{WB}}$ to improve scheduling performance. Furthermore, the range of actions per UE can be asymmetric with respect to $n_{k,\text{WB}}$.

3.3 Reward Design

As the scheduling metric is sum-TBS per slot, the reward should incarnate the allocated total TBS in a slot. Therefore, at each allocation step i , the allocated TBS per UE is first calculated, denoted as $\text{TBS}_{k,i}$ for UE k . Next, a temporary quantity p_i for allocation step i is computed as

$$p_i = \min \left(\sum_{k=1}^K \text{TBS}_{k,i}, \sum_{k=1}^K L_k \right) \Bigg/ \sum_{k=1}^K L_k \quad (3)$$

whose physical meaning is the normalized transmitted sum-rate. The ultimate reward at allocation step i , r_i , is the summation of p_j 's associated with all the allocation steps up to allocation step i , i.e.

$$r_i = \sum_{j=1}^i p_j \quad (4)$$

Since $\sum_{j=1}^i \sum_{k=1}^K \text{TBS}_{k,j} \leq \sum_{k=1}^K L_k$, the reward r_i always falls between 0 and 1, which facilitates the training of the DQN.

Table 2. Comparison of Online Computational Complexity

Algorithm	JADE	STAR
Number of scheduling metric calculation	MK^2	1
Sum complexity	MK^2	1
Sum complexity for the case of $K = 30$, $B = 270$, $M = 12$, $M_{\text{RB}} = 4$	$\mathcal{O}(1e4)$	1

3.4 Complexity Analysis

The most prominent advantages of the proposed STAR algorithm over JADE are its superior performance (to be shown later in this work) and its significantly reduced online computational complexity. Assuming each UE needs M RBs on average to send its payload, then the total computational complexity of JADE is around MK^2 [12], since it necessitates the calculations of the TBS from both ends of the BWP for each UE, and repetitions of the above procedure after each UE and resource allocation stage. In contrast, by using STAR, the computational complexity at each allocation step is only 1, since it directly determines which UE and how many RBs should be scheduled with only one-time TBS calculation, hence substantially reducing the complexity. Therefore, STAR can provide about MK^2 times, which can be up to four orders of magnitude, of complexity reduction as compared to JADE. A brief analysis and comparison of the online computational complexity of the two algorithms is provided in Table 2.

3.5 Training of DQN

In the DRL framework of this work, the agent is a DQN model composed of three fully-connected hidden layers where the number of neurons per layer is 1024, 256, and 128, respectively, in addition to an input layer and an output layer. The input size is $K(B + 1)$ which is the state dimension, and the output size is the cardinality of the action space, i.e. $5K$. The environment incorporates the wireless channel models as well as transmission and reception procedures based on the 3GPP 5G new radio (NR) standards such as [2–6], among others. In each training step, the following processes take place (as shown by Fig. 1): (1) The environment generates the current state s and passes it to the DRL agent; (2) The DRL agent generates an action a and feeds it back to the environment; (3) The environment yields the reward r and the next state \tilde{s} based on a , and outputs them to the DRL agent; (4) The sequence of state, action, reward, and next state, $[s, a, r, \tilde{s}]$, is used to train the DQN, i.e., the agent. Experience replay and ϕ greedy action selection are performed to facilitate the training [10].

4 Numerical Results

System-level simulations are carried out to evaluate the performance of the proposed algorithm. Table 3 lists the simulation settings, where the traffic models

Table 3. Simulation settings

Configuration	Value
Transmit power	23 dBm
Number of gNB antennas	4
Cell radius	250 m
UE distribution	Uniform
Number of antennas per UE	4
Number of UEs per gNB	5
Channel	EPA (Extended Pedestrian A model)
Numerology	30 kHz sub-carrier spacing, 100 MHz bandwidth
CSI feedback delay	1 slot
Traffic model	remoteDrivingDl (RDD), powerDist2 (PD2)

Table 4. Parameters for the traffic models used in simulations [1]

	remoteDrivingDl (RDD)	powerDist2 (PD2)
Delay threshold (ms)	1	1
Acceptable packet drop probability	0.0001%	0.0001%
Packet size (bits)	16664	2000

are remoteDrivingDl (RDD) and powerDist2 (PD2) which represent the downlink remote driving scenario, and the second type of power distribution grid fault and outage management [1], respectively, both of which belong to URLLC (Ultra-Reliable Low-Latency Communications), one of the three major 5G usage scenarios [1]. The total number of UEs in our simulations is $K = 5$, thus yielding 25 actions. Table 4 details the parameters for the traffic models used in our simulations. The training and inference processes of the proposed DRL is implemented in PyCharm. The relevant simulation parameter values of the DRL are given in Table 5.

The performance of STAR is compared against JADE and a random scheduling strategy, i.e., arbitrarily selecting which UE and how many RBs to be scheduled at each allocation step. The sum-TBS and resource utilization of these three algorithms are illustrated in Fig. 2, where the resource utilization is defined as the ratio of the number of consumed RBs to the total number of RBs, and the ratio of PD2 and RDD UEs is 1:4. Moreover, Fig. 3 and Fig. 4 show the performance with different UE ratios, using exactly the same offline-trained DRL model for the case in which the ratio of PD2 and RDD UEs is 1:4. The following key observations can be drawn from Figs. 2, 3 and 4:

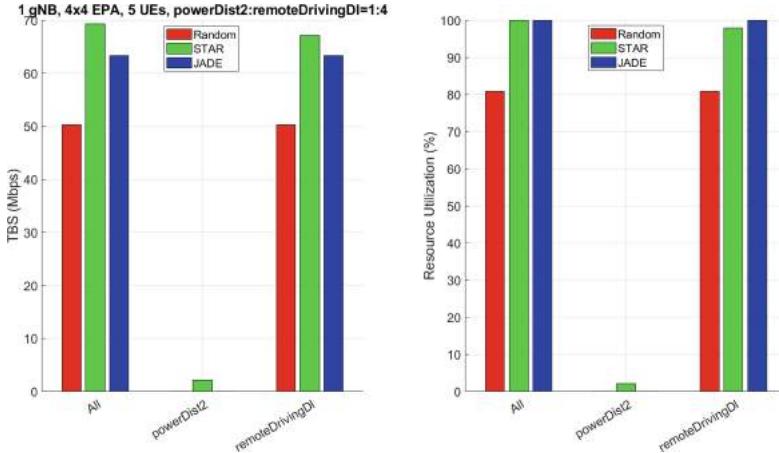


Fig. 2. Performance of JADE, STAR, and Random Scheduling Approaches. The Ratio of PowerDist2 and RemoteDrivingDl UEs is 1:4.

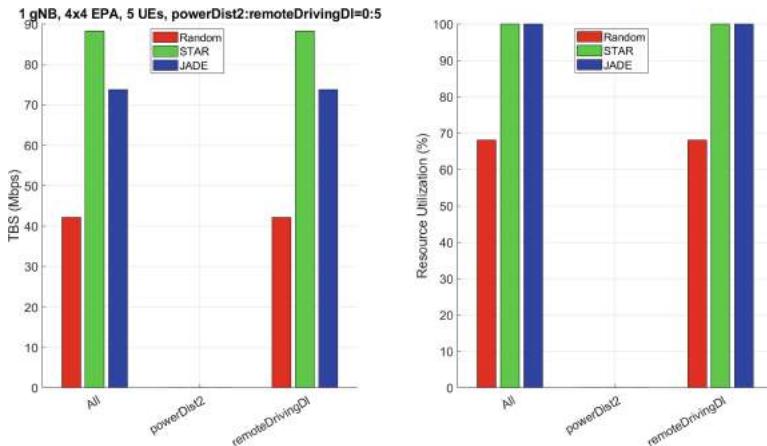


Fig. 3. Performance of JADE, STAR, and Random Scheduling Approaches. The Ratio of PowerDist2 and RemoteDrivingDl UEs is 0:5, And the RL Model used for the Simulation is with a Ratio of PowerDist2 and RemoteDrivingDl UEs of 1:4.

1) The TBS comparison in Fig. 2 unveils that STAR outperforms both JADE and random scheduling for both overall mixed and individual traffic types, demonstrating its superiority over the two schemes. Specifically, the TBS gain of STAR over JADE is about 10% for overall mixed traffic types, 6% for the RDD traffic, and virtually infinity for the PD2 traffic since JADE does not yield any TBS for PD2. Furthermore, the resource utilization comparison in Fig. 2 implies

Table 5. Simulation parameters of the DRL

Parameter	Value
Number of states	255
Number of actions	25
Number of hidden layers	3
Number of neurons per hidden layer	1024, 256, 128
Initial weight value	Normal initialization
Optimization algorithm	Adam
Activation function	ReLU
Epsilon decay rate	0.996
Learning rate	1e-6
Experience memory size	16400×536
Batch size	1024

that JADE assigns all RB resources to the RDD traffic, probably due to its significantly larger packet size (refer to Table 4) and the payload-exhaustiveness nature of JADE so that once an RDD UE is selected, it would consume RBs continuously until its payload is met. On the contrary, STAR is able to utilize the resources more efficiently and preserves some RBs for the PD2 UEs. Additionally, the random scheduling fails to make full use of the resources which leads to low TBS.

2) As indicated by Figs. 3 and 4, applying exactly the same offline-trained DRL model to different UE traffic ratios, STAR still yields the best TBS performance, which shows its robustness to traffic type distributions. For instance, when only RDD UEs are present, as shown in Fig. 3, the resource utilization for both STAR and JADE reaches 100%, but STAR manages to allocate the RBs more wisely so as to yield higher TBS than JADE. The random scheduling scheme produces the lowest TBS due to inefficient usage of the resources, which is consistent with the observation from Fig. 2. On the other hand, when only PD2 UEs exist (see Fig. 4), STAR can still yield the largest TBS as compared with the other two algorithms.

3) Combining the simulation results in Figs. 2, 3 and 4 and the complexity comparison in Table 2, it is evident that STAR surpasses JADE in terms of sum-TBS while having significantly lower online computational complexity, thus substantially outperforming JADE considering both system-level performance and online computational complexity. Therefore, it is advantageous to adopt STAR in practice to conduct joint multi-UE and resource scheduling with contiguous FDRA.

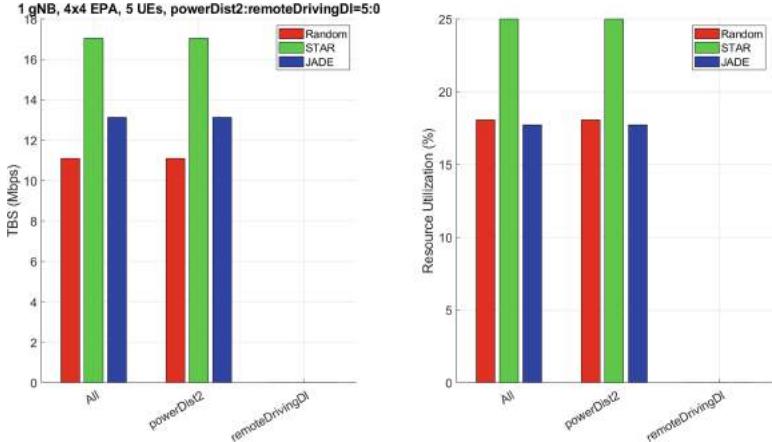


Fig. 4. Performance of JADE, STAR, and Random Scheduling Approaches. The Ratio of PowerDist2 and RemoteDrivingDI UEs is 5:0, And the RL Model used for the Simulation is with a Ratio of PowerDist2 and RemoteDrivingDI UEs of 1:4.

5 Conclusion and Discussion

We have proposed a DRL-based algorithm that jointly schedules UE and contiguous frequency-domain resources. In the proposed method, a DQN agent module is trained offline to determine which UE and how many RBs for that UE should be scheduled at each allocation step. We design the state space, action space, and reward function which are suitable for both the joint UE and resource allocation problem and DRL implementation. System-level simulations demonstrate that compared to a practical contiguous FDRA algorithm named JADE proposed in [12] which outperforms prior existing methods, the proposed algorithm enjoys both better performance and significantly lower online computational complexity. This work can be extended to the scenarios which may contain more UEs, more diverse traffic types, and/or other scheduling metrics.

While this work considers the sum-TBS as the scheduling metric, other metrics such as PF and modified largest weighted delay first (M-LWDF) can be utilized in practice as well, which may necessitate the re-design of the state and reward in the DRL framework.

Acknowledgment. The authors would like to thank the Next Generation and Standards Group (NGS) in Intel Corporation for their great support of this work.

References

1. 3GPP TR 38.824, V16.0.0. Study on physical layer enhancements for NR ultra-reliable and low latency case, March 2019
2. 3GPP TR 38.901, V16.1.0. Study on channel model for frequencies from 0.5 to 100 GHz, December 2019

3. 3GPP TS 38.211, V16.3.0. NR; Physical channels and modulation, September 2020
4. 3GPP TS 38.213, V16.3.0. NR; Physical layer procedures for control, September 2020
5. 3GPP TS 38.214, V16.3.0. NR; Physical layer procedures for data, September 2020
6. 3GPP TS 38.331, V16.1.0. NR; Radio resource control protocol specification, July 2020
7. He, X., Wang, K., Huang, H., Miyazaki, T., Wang, Y., Guo, S.: Green resource allocation based on deep reinforcement learning in content-centric IoT. *IEEE Trans. Emerg. Top. Comput.* **8**(3), 781–796 (2020)
8. Li, X., Alkhateeb, A.: Deep learning for direct hybrid precoding in millimeter wave massive MIMO systems. In: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, pp. 800–805 (2019)
9. Li, X., Alkhateeb, A., Tepedelenlioğlu, C.: Generative adversarial estimation of channel covariance in vehicular millimeter wave systems. In: 2018 52nd Asilomar Conference on Signals, Systems, and Computers, pp. 1572–1576 (2018)
10. Luong, N.C., et al.: Applications of deep reinforcement learning in communications and networking: a survey. *IEEE Commun. Surv. Tutor.* **21**(4), 3133–3174 (2019)
11. Puterman, M.L.: *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley, New York (2014)
12. Sun, S., Moon, S.: Practical scheduling algorithms with contiguous resource allocation for next-generation wireless systems. *IEEE Wirel. Commun. Lett.* **10**(4), 725–729 (2021)
13. Sun, S., Moon, S., Fwu, J.: Practical link adaptation algorithm with power density offsets for 5G uplink channels. *IEEE Wirel. Commun. Lett.* **9**(6), 851–855 (2020)
14. Tse, D.: Forward link multiuser diversity through proportional fair scheduling. Presentation at Bell Labs, August 1999
15. Tsiropoulos, E.E., Kapoukakis, A., Papavassiliou, S.: Uplink resource allocation in SC-FDMA wireless networks: a survey and taxonomy. *Comput. Netw.* **96**, 1–28 (2016)
16. Wong, I.C., Oteri, O.F., McCoy, J.W.: Resource allocation in multi data stream communication link. U.S. Patent 7 911 934, March 2011
17. Xu, Z., Wang, Y., Tang, J., Wang, J., Gursoy, M.C.: A deep reinforcement learning based framework for power-efficient resource allocation in cloud RANs. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–6 (2017)
18. Yan, H., Ashikhmin, A., Yang, H.: Optimally supporting IoT with cell-free massive MIMO. In: 2020 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2020)
19. Yan, H., Ashikhmin, A., Yang, H.: A scalable and energy efficient IoT system supported by cell-free massive MIMO. *IEEE Internet Things J.* (2021)
20. Yan, H., Lu, I.T.: BS-UE association and power allocation in heterogeneous massive MIMO systems. *IEEE Access* **8**, 184045–184060 (2020)
21. Ye, H., Li, G.Y., Juang, B.F.: Deep reinforcement learning based resource allocation for V2V communications. *IEEE Trans. Veh. Technol.* **68**(4), 3163–3173 (2019)
22. Zhang, C., Patras, P., Haddadi, H.: Deep learning in mobile and wireless networking: a survey. *IEEE Commun. Surv. Tutor.* **21**(3), 2224–2287 (2019)
23. Zhao, N., Liang, Y., Niyato, D., Pei, Y., Wu, M., Jiang, Y.: Deep reinforcement learning for user association and resource allocation in heterogeneous cellular networks. *IEEE Trans. Wirel. Commun.* **18**(11), 5141–5152 (2019)



A Novel Model for Enhancing Fact-Checking

Fatima T. AlKhawaldeh^(✉), Tommy Yuan, and Dimitar Kazakov

University of York, Deramore Lane, Heslington, York YO10 5GH, UK
`{ftma500, Tommy.yuan, dimitar.kazakov}@york.ac.uk`

Abstract. Fact-checking is a task to capture the relation between a claim and evidence (premise) to decide this claim’s truth. Detecting the factuality of claim, as in fake news, depending only on news knowledge, e.g., evidence text, is generally inadequate since fake news is intentionally written to mislead readers. Most of the previous models on this task rely on claim and evidence argument as input for their model, where sometimes the systems fail to detect the relation, particularly for ambiguous information. This study aims to improve fact-checking task by incorporating warrant as a bridge between the claim and the evidence, illustrating why this evidence supports this claim, i.e., If the warrant links between the claim and the evidence then the relation is supporting, if not it is either irrelevant or attacking, so warrants are applicable only for supporting the claim. To solve the problem of gap semantic between claim evidence pair, A model that can detect the relation based on existing extracted warrants from structured data is developed. For warrant selection, knowledge-based prediction and style-based prediction models are merged to capture more helpful information to infer which warrant represents the best bridges between claim and evidence. Picking a reasonable warrant can help alleviate the evidence ambiguity problem if the proper relation cannot be detected. Experimental results show that incorporating the best warrant to fact-checking model improves the performance of fact-checking.

Keywords: Fact-checking task · Cycle consistent adversarial network · Rebuttals and warrants

1 Introduction

Argument Mining aims mainly to predict the relationship between a claim and its relevant evidence: supporting, attacking or unverified (i.e., fact-checking) [1]. Fact-checking is a key mission of argument mining for many NLP tasks to detect support and attack relation between claim and premise [2, 3] such as entailment [4], document evaluation [5] and evidence detection in news domain [6, 7]. The work in [8] suggests that using Toulmin argument for an argumentation-based mechanism for reasoning in BDI agents enables the generation of claims and decides the level of confidence of these claims based on qualify function. Incorporating the warrant information helps solve the inability to distinguish between refuting, supporting, or non-relevant evidence [9].

For fact-checking, in some cases, the actual label could not be detected due to the absence of information that helps solve ambiguity problem, in other words, the implicit

explanation is necessary to support the claim by evidence. Previous works do not use warrant as complementary knowledge for improving fact-checking while in [10], Singh et al. use it as a hint for prediction the link between a claim and premise evidence detection. They use only correct warrant and select one warrant randomly from them to create the data as positive instances labelled (1), and for negative instances, they randomly choose premise for a claim with its correct warrant to rank a set of candidates' evidence and select best evidence piece to a particular claim. They depend only the correct warrant to identify the supporting evidence piece for a given claim. In this proposed model, warrants are ranked to select the best reasoning warrant to the argument instead of selecting the best evidence as to their work, in order to detect the relationship between claim and evidence. The warrant that leads to contradictory claim as a rebuttal from Toulmin argument is considered, in spite that warrants are given in the ARCC dataset. The model is also proposed to extract the relevant warrants and best of them to help make it extensible approach for external information, e.g., warrants extracting from Wikipedia or training this model on other different corpora. In simple words, Singh et al. [10] look for the best evidence piece from evidence candidates that support the claim. While the proposed model looking for the best warrant that is related to a claim and then identify the relation, support (if warrant), attack (if contradictory warrant) or non-relevant if the warrant is randomly selected and not related to the claim or may does not take a position towards the claim. The final difference is that Singh et al. used model [10] is Bi-LSTM, while in this work, a different model is developed as presented in Sect. 3.

Generally speaking, the differences between the existing works of the fact-checking and current presented approach is that fact detection of the previous model check only two-component, claim, and evidence while this model takes warrant into accounts as sometimes it is not clear what is the reason behind supporting this claim by its associated evidence. The justification for deciding whether this evidence has good reason to link (support) the claim to the evidence could be captured by another argument component. To alleviate this challenge, a deep architecture for the fact-checking task is modelled considering Toulmin arguments: warrant and rebuttals as in [11, 12]. Freeman and Toulmin proposed a model of arguments in which the premise supports the claim on behalf of warrants, which is an inference rule. The Toulmin model composition of an argument is the Premise (since) Warrants (then) Claim unless rebuttals. This work incorporates warrant or alternative warrant as complementary knowledge that bridge the gap between claim and evidence explaining why a particular claim follows from its evidence, so warrants are applicable for supporting the claim, and alternative warrants are applicable for attacking claim.

From a set of candidates warrant, this work applies a model to infer which warrant is best bridges the reasoning gap from multiple warrants between a given claim and evidence. To achieve that, the proposed model merges style-based and knowledge-based models. Here, in the proposed best warrant selection, the only related work to this model is warrant ranking in Singh et al., 2020 [13], the difference between this work's best warrant model and their model: they rely on crowd workers to select the more relevant and best warrants, while this work automated the ranking models instead of using manual

methods. An example of a given claim, evidence and five candidates warrant collected from with its ranking [13], is shown below:

- An example: Suppose the following example, an instance of five candidate warrants (W1–W5) for a given claim and evidence pair, the ranking warrants are 3, 4, 2, 1, 5, where W3 can be considered better reasoning from evidence to claim.
- The claim: “There is no clear division between the force required to knock a person out and the force likely to kill a person”.
- The evidence: “The first boxing rules, called the Boughton’s rules, were introduced by champion jack brought on in 1743 to protect fighters in the ring where deaths sometimes occurred”
- Warrant 1: “Is being a force of knocking person where death will sometimes be Fight occurred.”
- Warrant 2: “The fighting between two persons in-ring should have some common rules to where to knock a person in head and face.”
- Warrant 3: “In boxing sports, many strict rules are followed to protect the fighters from death and heavy injurious.”
- Warrant 4: “The force required to ring death occurred the protect fighters.”
- Warrant 5: “The boxer start fights in the ring if death occurs in it.”

For the knowledge-based model, this work depends on a deep neural network for effective text representations, where the word is represented as an embedding vector to learn semantic representations for the pair of text and then to reach the matching score. For the style-based model, it employs a cycle consistent adversarial network (CycleGAN) model. It uses the generative variational autoencoder [3, 14] as it is widely used for style transfer generation and CycleGAN architectures. Meanwhile, a discriminator consisting of CNN estimates the probability that the transferred text comes from the target semantic domain. It uses CycleGAN, where it has two conditional generators and two discriminators. For each generator, the model uses variational autoencoder with latent space [15] as in Fig. 1 where c is Structured latent space (style code), and z is Unstructured latent space (remaining information). Each generator conditioned on the output from two GRU encoders, the first encoder to encode the other domain’s style, and the second encoder is for other content encodings. CycleGAN has two generators combined with two discriminators to learn two bidirectional mappings for data input; it is initially developed for the image to image transformation to learn a transformation between image domains [16, 17]. The successes of the cycle consistent adversarial network (CycleGAN) in image domain transformation and semantic matching relation as using CycleGAN combined with the transformer network [18], encouraged us to suggest an argumentative relation identification task based on the style transfer.

Main contributions are as follows:

- 1) The proposed fact-checking model incorporates the warrant and rebuttal information to improve the performance of labelling the factuality of claim.
- 2) The selection of a good warrant and identify the best warrant for all claim and evidence pairs provides more guidance for the fact-checking task. So, the effect of picking the best warrant is remarkable.

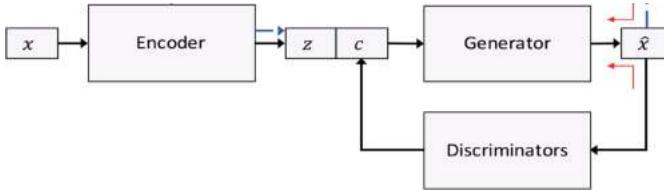


Fig. 1. Variational autoencoder with latent space [15]

- 3) Both the semantic representation and the semantic transformation used to transfer the claim and evidence style to the warrant style to check their similarity or dissimilarity by a novel framework contribute to the best warrant selection.
- 4) For the fact-checking model, combining linguistic features such as sentiment and subjectivity helps the model decide. As it supposes supporting the claim, e.g., the sentiment should be the same for all inputs.

2 Related Work

There are several research concerns by fact-checking detection like depending on expressive features as TFIDF features [19] and other features. Karadzhov et al. [20] depend on ground truth as a credible source like google engine and applied LSTMs set on retrieved results to enrich SVMs and multilayer perceptron's to detect the factuality. Evidence is extracted by searching from trusted websites to verify news based on claim queries method. They develop a dataset that consists of 992 sets of tweets are used for experiments. Despite satisfactory results by following question answering which needs generating queries and selects the best snippet than the best sentences, it is extremely complicated and needs too much processing to get the final factuality label. The system has the benefit of not depending on highly engineered features. In this automatic system, the information they gain comes from the web, the accuracy may be affected according to how much the retrieved snippet is relevant and to which degree the sources are trusted.

Recurrent/Recursive Neural Networks (RNNs) are used to represent sequential posts and user engagements [21–24], in these research tweet propagation data. Convolutional Neural Networks (CNNs) to capture local features of texts and images [25] are applied to focus on unigram word features, or both [26]. This study takes images into account to check veracity. Combining images processing with texts needs sperate tasks and mixed datasets. Another risky if the image is true and the text is true but not related. Generative Adversarial Networks (GANs) have also been used and extended to obtain a “general feature set” for fake news across events to achieve fake news early detection [27].

Ma et al. [21] represented fake news propagation by RNN based on a bottom-up and a top-down tree-structured neural networks by focusing on user properties and profiles information. The risky in this system is to which extent the user profile is real and not fake, or there are different reasons behind creating this profile and opposing opinions or comment as “fake deny” or “fake support”. Ruchansky et al. [22] developed CSI model specification with three components: Capture module based on LSTM to get textual information of the pattern of temporal engagement to an article, while the Score module

extracts source characteristic for all users, the combination is done between article representation which comes from the first module and user information representation that comes from the second module, they are combined in integrating module to classify the fakes news.

RNN model is used to extract the relationship between creators of news and subject [23]. LSTM is used to extract the representation of temporal textual characteristics (time-series event) of rumour to classify the rumour on Twitter early. Even this system can learn without training heavy manual features hidden representations incapable of detecting dynamic structures for a long time [24]. Multiple convolutional layers are implemented to merge the input representation from both images and texts. This model obtains good results, but it needs a huge data to train. Liu and Wu [26] proposed an early detection system to directly catch the false claims after posting by training the merged CNN and RNN.

It is noticed that DNN models are mainly used with good results, but some of these systems have computational limitation like in [20], where retrieving evidence to compare with sometimes taking a long time, especially filtering process. Some of these systems need a lot and continuity observing changes in a sequence of posts in user additions, and they only trained for only supervised data [21–24]. Wang et al. [27] extract textual and visual features to train the models there is a risk that images features have more transferability than texts and the training and experiment have been done on imbalance Twitter dataset. Despite Ruchansky et al.'s [22] promising results, but could not be reliable since there is a lack of ground truth information about users where there is a possibility to publish fake data about them and have the problem predict unobserved users due to depending on user features training. Training on small dataset makes it difficult for CNN to train and detect the meaningful patterns in texts, and CNN does not deal with long dependencies sequences.

The link between a claim and candidate evidence is detected by the fact-checking and argumentative relation identification tasks. Recently more work has been done for detecting for finding the relationship between a claim and premise using both supervised and unsupervised models [2, 3, 5]. Nguyen and Litman [28] use adjacent context features and the main topic. Kurabayashi et al. [29] extract argumentative flow as an essential feature to capture the link for argumentative relation identification. Discourse structures where discourse markers are the features rely on detecting the relations [5]. Peldszus and Stede [1] focused on the tasks of discovering the structure of the arguments such as argument structure parsing considering only the relevant argumentative component of the text. Concarascu and Toni [2] applied deep learning model by classifying pairs of text pieces to identify the relation between them. Dataset for identifying context-dependent evidence is built-in [30]. For incorporating complementary knowledge like implicit warrant, Boltuzic & Šnajder [31] and (Habernal et al. [32] show that it is beneficial in linking the premise and the claim as a hint for prediction also it helps for unseen data.

For news domain, content-based models mostly applied, particularly knowledge-based models which use complementary sources for fact-checking by identifying support and attack relation [6, 7]. Other models do not rely on complementary information and consider the style of claim: style-based, such as focusing on subjectivity analysis and whether the writing style is manipulated [33, 34]. For text style transfer, Hu et al. and Shen

et al. [15, 35] have developed an encoder-decoder architecture with style discriminators, such as a binary CNN based discriminator where the encoder has the role in learning a style independent knowledge representation while decoder outputs the transferred sentence considering both the knowledge representation and the desired style (target style).

3 The Proposed Fact-Predictor Architecture

3.1 Key Idea

This proposed model's main aim is to classify the relationship between a claim and candidate evidence: supporting evidence, attacking evidence or non-supporting (non-relevant) evidence. To enhance the fact-checking task, this model incorporates the warrant (or alternative warrant which acts as rebuttal), which helps decide the factuality of claim and uses multi-channel combined with multi-head attention for fact prediction.

The proposed model architecture has two components; as shown in Fig. 2, First component is Hierarchical Reinforcement Learning (HRL) approach: A high-level policy for plausible Warrant extraction (Sect. 3.2); low-level policy for best warrant picking (Sect. 3.3). The second component is Fact-predictor (Sect. 3.4), which is also used to provide a reward to guide both a high-level and a low-level policy.

To achieve this, HRL is applied, where high-level policy decides whether a warrant is plausible to the claim evidence pair or not, the low-level policy is trained to select the best warrant by merging two policies. The model merges two models' outputs, knowledge and style-based prediction models considering warrant as a complementary source. These models' inputs are triple of (claim, warrant, evidence) as (c, w, e). For knowledge-based prediction, capsule and BiGRU networks are applied to better represent syntactic and semantic information. For style-based prediction, feature guided conditioned cycle GAN via VAE is applied to transfer the style of a text to the desired style then check the style matching. The final component in our model is fact-predictor used to decide the factuality of claim. Also, it provides a reward to detect the relationship between a claim and evidence and guide all policies.

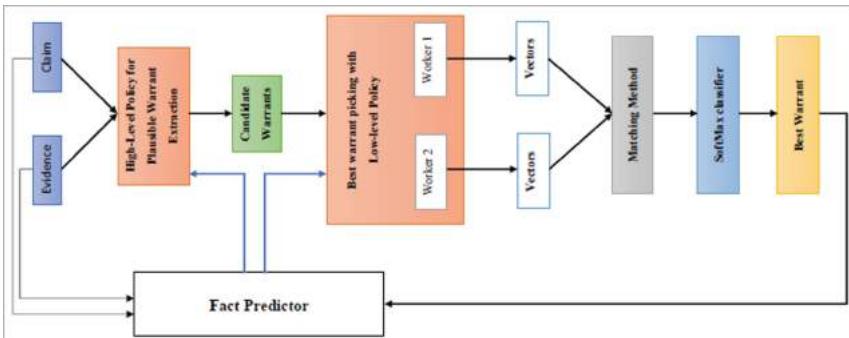


Fig. 2. The proposed model architecture

3.2 A High-Level Policy for Plausible Warrant Extraction

The high-level policy's main goal is extracting the most relevant warrants that are more plausible for the claim evidence pair and discard the irrelevant ones. After the warrant is selected as candidate relevant warrant, it is passed as subgoals to the low-level policy for best warrant decision.

Policy: This policy depends on effective text representations after the embedding vector represents each word with an attention mechanism. For each time step, the input is (c, w, e) to each claim evidence pair. Next, a deep neural network is applied: Bidirectional Long Gated Recurrent Unit (BiGRU) combined with the attention mechanism to capture crucial information from both directions. To encode the inputs and obtain the contextual information, the model uses BiGRU which can efficiently use past features and future features and summarise these words information from both directions (forward and backward) as in Eqs. 13.

$$\vec{h}_i = \overrightarrow{\text{GRU}}_{(c_i)}; \quad i \in [1, C] \quad (1)$$

$$\hat{h}_i = \overleftarrow{\text{GRU}}_{(c_i)}; \quad i \in [C, 1] \quad (2)$$

$$h_i = \vec{h}_i \oplus \hat{h}_i \quad (3)$$

To compute the attention weight a_i between each claim hidden state h_i and the warrant representation w , the similarity between the hidden states of the claim and the representation of warrant, the following equations are applied [4–6]:

$$h_w^p = \sum_{i=1}^n h_w^i / n \quad (4)$$

$$m_i = \tanh(W_c \cdot [h_i; h_w^p] + b_c) \quad (5)$$

$$a_i = \text{softmax}(m_i) = \frac{\exp(m_i)}{\sum_{t=1}^C \exp(m_t)} \quad (6)$$

h_w^p is pooling vector for the warrant. The claim representation c can be got based on the attention vectors a_i by:

$$cr = \sum_{i=1}^C \alpha_i \cdot h_i \quad (7)$$

For evidence representation er , the above equations are used by only replacing claim c by evidence e . To rank multiple warrants for a given claim and evidence pair, this model feeds both claim and evidence representation to a SoftMax classifier where the highest probability stands for the best warrant to fill the gap between the claim and the evidence. Where \oplus represent the connection of vectors, claim representation cr and evidence representation er

$$score = \text{softmax}(w.[cr \oplus er] + b) \quad (8)$$

It is as guidance to rank the candidate's warrants [$w_1; w_2 \dots w_n$] and select the good one whose reasoning is better than others for the claim and the evidence to classify the relationship between a claim and evidence.

The high-level policy utilizes a reward to select the warrants to guide warrant extraction over the warrants sequence. The actions taken by this policy depends on the score result, Eq. 8, which is a conditional probability of binary classification while the state at each time claims evidence pair. More details about the state, action and reward of the high-level policy are described as follows.

State: the state composed of three parts, claim and its evidence in addition to a candidate warrant from the data. Police use this information to decide either select the candidate warrant as relevant or not.

Action: *the policy samples action $a_i, j \in \{0, 1\}$ by the conditional probability, as defined in score Eq. 8*

Reward: After the policy has taken action, this action should be rewarded b cosine similarity between the hidden states of the selected warrant and the word embedding of the claim evidence pair.

3.3 Best Warrant Picking with Low-Level Policy

Given a relevant warrant sequence $\{w_1, 1, \dots, w_n\}$ by the high-level policy, low-level policy π_l aims to select the best warrant with better reasoning and discard fewer ones. The low-level policy has two workers. Each of them trained independently; they help the low-level policy learn to select the best warrant among warrant sequence (taking action) by considering different points of view, the semantic representation, and the style representation as worker one and worker two respectively. The low-level policy considers all outputs from both workers and passes them to the SoftMax output layer to produce probabilities over warrants. For each warrant, the conditional probability by SoftMax keeps until all candidates are processed then the highest probability is assigned for the best warrant as a decision for the policy. More details are defined in Sect. 3.4. The state, and reward of low-level policy are defined as follows.

State: are the information of relevant warrant sequences that comes from the high-level policy, also, to claim evidence pair for deciding to select the best warrant.

Action: this policy adopts SoftMax function to decide the best warrant based on conditional probability results.

Reward: After the policy has taken action, this action should be rewarded entailment metric between the hidden states of the selected warrant and the word embedding of the claim evidence pair.

Worker1: Knowledge-Based Prediction (Semantic Representation). In this paper, a capsule network model incorporating BiGRU is proposed, Siamese BiGRU capsule networks. This model consists of two parts, capsule module and Bi GRU module. Initially, a claim is represented by a word embedding where words from the vocabulary are mapped to vectors. In this model, the pre-trained word vector GloVe is employed as inputs to the model as follow.

$$x_{in} = e_w^c w_{in}, n \in [1, N] \quad (9)$$

BiGRU is used to capture long-distance dependencies within a sentence and proved its effectiveness to encode sentence representation. It captures the information from both directions left and the right context then the word representation is the concatenation for them.

$$\overrightarrow{h}_{in} = \overrightarrow{GRU}(x_{in}), n \in [1, N] \quad (10)$$

$$\overleftarrow{h}_{in} = \overleftarrow{GRU}(x_{in}), n \in [N, 1] \quad (11)$$

$$h_{in} = \overrightarrow{h}_{in} + \overleftarrow{h}_{in}, n \in [1, N] \quad (12)$$

Capsule Network: Hinton et al. and Sabour et al. [36, 37] suggest Capsule model instead of the single neural node as in CNN models; they use neuron vector for input and output layer with dynamic routing algorithm instead of pooling operations. It is used for understanding the spatial information and learn contextual information of text in different tasks. For example, question answering [38], word segmentation [39] and extract the global semantic features of different categories [40], sentiment analysis [41–43], cross-domain [43, 44], sarcasm detection [45], propaganda detection [46].

According to Gao et al. [47] capsule network is robust to extract a richer representation of a text and other significant features such as word position, the semantic and syntactic structure. This proposed model applies Gao et al. [47] equations to obtain the capsule network's output and considers other output of BiGRU. The outputs of capsules networks are achieved in Eqs. 13, 14 and 15.

$$\hat{u}_{o|i} = w_{io} h_{in} \quad (13)$$

$$S_{out} = \sum_{i=1}^m c_{io} \hat{u}_{o|i} \quad (14)$$

$$c_{io} = \frac{\exp(b_{io})}{\sum_k \exp(b_{ik})} \quad (15)$$

Where c_{io} is the coupling coefficient, is determined by the dynamic routing method and S_{out} is vector representation. The activation of capsule network output is calculated by the nonlinear function (Squash) for normalization purpose, as shown in Eq. 16.

$$v_{out} = \frac{\|S_{out}\|^2}{1 + \|S_{out}\|^2} \frac{S_{out}}{\|S_{out}\|} \quad (16)$$

Where v_{out} is the output vector of the capsule network. For c is v_{outc} , for v_{outw} and for e v_{oute} The dynamic routing method [43] is shown below:

for all capsule i in layer l and j in $l+1$:

initial: $b_{ij} \leftarrow 0$.

for iterations do

$$c_{ij} \leftarrow \text{soft max}(b_y) \quad (17)$$

$$s_j \leftarrow \sum_i c_{ij} u_{j|i} \quad (18)$$

$$v_j \leftarrow squash(s_j) \quad (19)$$

for all capsule i in layer l and j in l + 1: $b_{ij} \leftarrow b_{ij} + u_{j|i} \cdot v_j$.

return v_j .

After generating the vector representation of each claim, warrant and evidence, All the resulting vectors are concatenated and fed to a SoftMax classifier to predict the relation between C and P (to express labels 0, 1, 2).

Worker 2: Style Based Prediction (Semantic Transformation). The motivation behind using the style-based model matches the transferred text Qx of a target text style y with its original style x to check the semantic distance between them.

This model has three texts, c , w , e . First, it checks the style of claim toward the warrant, two texts c and w that respectively, transfer c to Qw and w to Qc then match each transferred text to the original one: c with qc and w with qw via Manhattan distance. The same thing of evidence, only replace c with e . It averages all Manhattan outputs the maximum average.

For both claim and evidence: If the original features of warrant used to generate transferred text, predicted features are close to them. The generator network combines the same z from claim or evidence and c from warrant and vice versa, to generate a text that satisfies the new constraints encoded in a specific style and preserving the knowledge. A discriminator consisting of CNN estimates the probability that the transferred text comes from the target semantic style domain and determine how the generated text acceptable. The generators and discriminators are adversarially trained with backpropagation.

After deciding the best warrant, the model moves to the next step to identify the accepted relation based on semantic relation.

The Policy: The Hybrid Model of Semantic Transformation and Representation. The relation is detected by merging vector representations from both style-based models and knowledge-based models using Concatenation, Elementwise product and elementwise difference matching methods. All the outputs are concatenated and fed to a SoftMax classifier. Table 1 shows the vectors representations of matching methods for both the style-based vectors and the knowledge-based vectors.

Table 1. Vectors representations of matching methods

Matching method	The style-based vectors	The knowledge-based vectors
Vectors concatenation	$(c + qc) (w + qw) (e + qe) (w1 + qw1)$	$(v_{outc} + v_{outw} + v_{oute})$
Vectors elementwise product	$(c * qc) (w * qw) (e * qe) (w1 * qw1)$	$(v_{outc} * v_{outw} * v_{oute})$
Elementwise difference	$(cqc) (wqw) (eqe) (w1qw1)$	$(v_{outc} - v_{outw} - v_{oute})$

Fact Predictor: Multi-channel Multi-head Attention Based BLGRU Siamese Network

Word Embedding Layer: All inputs (claim, warrant, evidence) are fed to the input layer. Each input is connected to the embedding layer, which builds word embeddings using Elmo, GloVe and fastText. In this work, to build word embeddings; Elmo, GloVe and fastText, that generates a word vector table, are used. For each input, all word vector of the word embeddings that are generated by Elmo, GloVe and fastText are concatenated as a matrix that is finally fed as inputs C, W and E claim, warrant and evidence respectively to the BiGRU layer.

Word Encoder Layer: Each word in each input C, W, and E that are words are represented using multi-channel of word embedding. WORD ENCODER LAYER creates a new representation for each word by summarizing contextual information from forward and backward directions using BGRU from both the directions in a comment. To obtain hidden state representation ht for each word for the whole input of (C, W, or E), forward hidden state and backward hidden state are concatenated for each word, and all of them are represented as H for each input.

Multi-head Attention Layer: For claim fact detection, each part in each input, claim, warrant, and evidence have specific part with variant role from different factors, so, this model focuses on them by applying multiple heads of attention then represents the semantics of the three inputs. After the whole hidden states are fed to the attention layer as Eq. 9, the whole representation semantics of each input is represented as Eq. 10, and Eq. 11 where $Wk1$ and $Wk2$ are parameters, Z is weight vector. Final input representation is M. Other feature vectors are merged to the final input representation, Linguistic features F: the sentiment feature vector and other Linguistic Inquiry and Word Count (LIWC) [48] features such as subjective, number, Swear, Negation and speculation expressions. Cnew is a new sentence representation that concatenated the input representation with the Linguistic features as Eq. 12. For all output Cnew for all inputs, the model merges them to generate a new representation V that pass to SoftMax layer to detect the label of fact-checking output as Eq. 13. the label with the highest probability stands for the predicted fact output. The predicted label is used as a reward for HRL policy.

$$A = \tanh(Wk1HT) \quad (20)$$

$$B = \text{SoftMax}(Wk2A) \quad (21)$$

$$C = BH \quad (22)$$

$$C_{\text{new}} = C + F \quad (23)$$

$$\text{Label} = \text{SoftMax}(WvV + bV) \quad (24)$$

4 Experiments

4.1 Dataset

For warrant selection: the corpus of ranked warrants from Singh et al., 2020 [13] are utilized, since it is the only available data for ranking warrants and select the best from a set of warrant candidates, this data is annotated for warrants preference learning, where a list of ranked warrants according to how well they connect a particular claim with a given piece of evidence. They labelled warrants for 100 claim-evidence pair, from top rank (high score) to bottom (low score) ranked warrants, each pair are annotated with five warrants.

For the fact-checking task, ARCC data¹ is used, stand for Argument Reasoning Comprehension Corpus from news comments [15] that is the build for SemEval task 2018 [9] by Habernal et al. [32]. The argument reasoning comprehension task is picking the right implicit warrant from two choices provided with an argument, a claim, and a premise. For the evidence detection task, Singh et al. 2019 [10] modify this data to be more appropriate to decide the relation between claim and evidence, the relation label is either support or Non-relevant. They label the datasets given the tuples of (Premise, Claim and Correct warrant) as a positive label, e.g. 1 and the tuples of (random Premise, Claim and correct warrant) as negative label 0(non-supporting). Singh et al. [10] paper present a deep learning-based approach for evidence detection by incorporating “correct warrant” as a bridge between the claim and the evidence to decide the evidence supports the claim. The positive and negative labels of their data for their experiments using only the correct warrants, are as follow

```
{claim, correct warrant, correct premise, label 1}
{claim, correct warrant, random premise, label 0}
```

The proposed model’s goal is to classify the relationship between a claim and the candidate evidence (i.e. whether it supports, attacks or it is irrelevant), so it considers both: correct warrant that explains why premise supports the claim and the alternative warrant that leads to contradictory claim as rebuttal rather than a warrant. Also, the random warrant is selected as non-relevant. The data instances as follow:

```
{claim, correct warrant, correct premise, label 1},
{claim, random warrant (non-relevant), correct premise, label 0},
{claim, attack warrant (rebuttal), correct premise, label 2}.
```

4.2 Settings

The embedding matrix is utilized with word2vec embeddings. Models were implemented in Keras, using TensorFlow for implementing. The proposed model used 20% of the data as test data list of hyperparameters used to train the neural architectures is presented in Table 2.

¹ <https://github.com/UKPLab/argumentreasoning-comprehension-task/>.

Table 2. Hyperparameters used to train the neural architectures

Hyperparameter	Value
Batch size	32
Embedding size	300
GRU cell size	128
GRU dropout	0.2
Optimizer	Adam
Learning rate	0.001
the number of route iterations	3
regularization constant of the dropout layer	0.2
the number of capsules	400

4.3 Results and Discussions

For warrant selection, results are evaluated by the normalized version of Mean Reciprocal Rank, Mean Quantile (MQ) score [49] which measures the correct ranks among all candidates warrants, and obtain 0.73 MQ score where MQ is mean quantile as well as the quantile ranging from 0 to 1.

More reasonable selected warrants increase the performance of this model. It is observed that the proposed model sometimes mislabelled the relation when the warrant has noise information or less relevant. There is no comparison between the proposed fact predictor model with other works in experiments since no previous work considering the warrant and rebuttal for fact-checking has been applied. The performance is evaluated using Accuracy which calculated as Eq. (25):

$$\text{Accuracy} = T/C \quad (25)$$

T is the number of correctly classified labels, and C is the number of true labels. The modified data has 1,210 instances as training data, and 444 instances as test data for each relation, i.e. the model collects only correct warrants for support relation, alternative warrant (rebuttal) for attack relation and randomly warrant for no-relevant relation.

The results of experiments presented in Table 2 show that selecting the best warrant from a set of correct warrants, instead of a randomly selected correct warrant, increases the model's performance by providing more information to decide the final relation label. In the training data, the best accuracy reached to 81.69. It is noticed that the sentiment, negation and other style information help to alleviate the ambiguity problem and make it clearer to capture the correct relation. For example, if the warrant (or alternative warrant) has the same polarity with claim and evidence, it is more likely to detect support relation, while attack relation could be detected when the claim's polarity is the opposite. Negation words help to detect the attack relation. The non-relevant label mostly occurs when the topic of claim and evidence is far from the warrant. Table 3 shows the results for evidence detection task, focusing only on the correct warrant and

Table 4 shows the results considering warrant, rebuttal and non-relevant information as the bridge between claim and evidence.

Table 3. Performance of evidence detection models in [10], they only use correct warrant

Model	Accuracy
Bidirectional LSTM model without a warrant [10]	72.71
Bidirectional LSTM model with the correct warrant [10]	76.74

Table 4. Performance of fact-checking in the proposed model and the impact of correcting the best warrant from multiple correct warrants, in addition to alternative warrant (rebuttal) or other irrelevant information

Fact predictor without warrant rebuttal, or non-relevant information (given only claim-evidence pair)	73.95
Random correct warrant aware -fact predictor	79.21
Best correct warrant aware- fact predictor	81.69

5 Conclusion and Future Work

This work has presented a novel model for the fact-checking task for the news domain to identify the relation between claim evidence if supported, not supported or attacked. The model incorporates warrant as additional knowledge and considers the style features and semantic representation features to select the best warrant. The results show that selecting better reasoning (warrant) given a claim, and evidence can help identify the correct relation. The results of the model look more promising than treat them separately. For future work, we will consider more arguments of the Toulmin model like baking and qualifier, in addition, to apply other models for style transfer generation such as transformer network provided with more conditioned features.

References

1. Peldszus, A., Stede, M.: Joint prediction in MST-style discourse parsing for argumentation mining. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, no. September, pp. 938–948 (2015). <https://doi.org/10.18653/v1/d15-1110>
2. Cocarascu, O., Toni, F.: Identifying attack and support argumentative relations using deep learning. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, vol. September 7, pp. 1374–1379 (2017). <https://doi.org/10.18653/v1/d17-1144>
3. Lippi, M., Torroni, P.: Context-independent claim detection for argument mining. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligent (IJCAI 2015), vol. January, pp. 185–191 (2015)

4. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 632–642 (2015). <https://doi.org/10.18653/v1/d15-1075>
5. Stab, C., Gurevych, I.: Identifying argumentative discourse structures in persuasive essays. In: 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference, EMNLP 2014, no. October, pp. 46–56 (2014). <https://doi.org/10.3115/v1/d14-1006>
6. Magdy, A., Wanas, N.: Web-based statistical fact checking of textual documents. In: Proceedings of the 2nd International Workshop on Search and Mining User-Generated Contents, no. October, pp. 103–109 (2010). <https://doi.org/10.1145/1871985.1872002>
7. Wu, Y., Agarwal, P.K., Li, C., Yang, J., Yu, C.: Toward computational fact-checking. Proc. VLDB Endow. 7(7), 589–600 (2014). <https://doi.org/10.14778/2732286.2732295>
8. de Oliveira, V., Gabriel, A., Panisson, R., Bordini, D., Adamatti, C., Billa, C.Z.: Reasoning in BDI agents using Toulmin’s argumentation model. Theor. Comput. Sci. **805**, 76–91 (2020). <https://doi.org/10.1016/j.tcs.2019.10.026>
9. Habernal, I., Wachsmuth, H., Gurevych, I., Stein, B.: SemEval-2018 task 12: the argument reasoning comprehension task. In: Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), vol. June, pp. 763–772 (2018). <https://doi.org/10.18653/v1/s18-1121>
10. Singh, K., Reisert, P., Inoue, N., Kavumba, P., Inui, K.: Improving evidence detection by leveraging warrants. In: Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), no. November, pp. 57–62 (2019). <https://doi.org/10.18653/v1/d19-6610>
11. Freeman, J.: Argument strength, the toulmin model, and ampliative probability. In: van Eemeren, F.H., Garssen, B. (eds.) Pondering on Problems of Argumentation, pp. 191–205. Springer, Dordrecht (2009). https://doi.org/10.1007/978-1-4020-9165-0_14
12. Toulmin, S.E.: The Uses of Argument. Cambridge University Press, Cambridge (1958)
13. Singh, K., Simpson, E., Reisert, P., Gurevych, I., Inui, K.: Ranking warrants with pairwise preference learning. In: Proceedings of the 26th Annual Meeting of the Natural Language Processing Society (March 2020), no. C, pp. 776–779 (2020). https://www.anlp.jp/proceedings/annual_meeting/2020/pdf_dir/P3-34.pdf
14. Mueller, J., Gifford, D., Jaakkola, T.: Sequence to better sequence: continuous revision of combinatorial structures. In: Proceedings of the 34th International Conference on Machine Learning, ICML, vol. 5, no. 1, pp. 3900–3916 (2017)
15. Hu, Z., Yang, Z., Liang, X., Salakhutdinov, R., Xing, E.P.: Toward controlled generation of text. In: 34th International Conference on Machine Learning, ICML 2017, vol. 4, no. PMLR 70, pp. 2503–2513 (2017)
16. Knyaz, V.A., Kniaz, V.V., Remondino, F.: Image-to-voxel model translation with conditional adversarial networks. In: Leal-Taixé, L., Roth, S. (eds.) ECCV 2018. LNCS, vol. 11129, pp. 601–618. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11009-3_37
17. Engin, D., Genç, A., Ekenel, H.K.: Cycle-Dehaze: enhanced CycleGAN for single image dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, pp. 938–946 (2018). <https://doi.org/10.1109/CVPRW.2018.00127>
18. Zhang, S., Tan, H., Chen, L., Lv, B.: Enhanced text matching based on semantic transformation. IEEE Access **8**(February), 30897–30904 (2020). <https://doi.org/10.1109/ACCESS.2020.2973206>
19. Karadzhov, G., Gencheva, P., Nakov, P., Koychev, I.: We built a fake news & click-bait filter: what happened next will blow your mind!. In: Proceedings of Recent Advances in Natural Language Processing, vol. September, pp. 334–343 (2017). https://doi.org/10.26615/978-954-452-049-6_045

20. Karadzhov, G., Nakov, P., Màrquez, L., Barrón-Cedeño, A., Koychev, I.: Fully automated fact checking using external sources. In: International Conference on Recent Advances in Natural Language Processing, RANLP, vol. 2017-Septe, pp. 344–353 (2017). <https://doi.org/10.26615/978-954-452-049-6-046>
21. Ma, J., Gao, W., Wong, K.: Rumor detection on twitter with tree-structured recursive neural networks. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers), pp. 1980–1989 (2018). <https://doi.org/10.18653/v1/P18-1184>
22. Ruchansky, N., Seo, S., Liu, Y.: CSI: a hybrid deep model for fake news detection. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, vol. Part F1318, no. November, pp. 797–806 (2017). <https://doi.org/10.1145/3132847.3132877>
23. Zhang, J., Dong, B., Yu, P.S.: FAKEDETECTOR: effective fake news detection with deep diffusive neural network. In: Proceedings of the International Conference on Data Engineering, vol. April, pp. 1826–1829 (2020). <https://doi.org/10.1109/ICDE48307.2020.00180>
24. Ma, J., et al.: Detecting rumors from microblogs with recurrent neural networks. In: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI 2016), pp. 3818–3824 (2016). https://ink.library.smu.edu.sg/sis_research/4630
25. Yang, Y., et al.: TI-CNN: convolutional neural networks for fake news detection. CoRR, vol. abs/1806.0 (2018). <http://dblp.uni-trier.de/db/journals/corr/corr1806.html#abs-1806-00749>
26. Liu, Y., Wu, Y.F.B.: Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks. In: 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, pp. 354–361 (2018). <http://dblp.uni-trier.de/db/conf/aaai/aaai2018.html#LiuW18>
27. Wang, Y., et al.: EANN: event adversarial neural networks for multi-modal fake news detection. In: Proceedings of The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, vol. Article 4, pp. 849–857 (2018). <https://doi.org/10.1145/3219819.3219903>
28. Nguyen, H.V., Litman, D.J.: Context-aware argumentative relation mining. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL, vol. 1: Long Paper, no. August, pp. 1127–1137 (2016). <https://doi.org/10.18653/v1/p16-1107>
29. Kurabayashi, T., Reisert, P., Inoue, N., Inui, K.: Towards exploiting argumentative context for argumentative relation identification. In: Proceedings of the 24th Annual Conference of the Society of Language Processing, March 2018, no. C, pp. 284–287 (2018). http://anlp.jp/proceedings/annual_meeting/2018/pdf_dir/A2-4.pdf. <https://www.google.com/search?q=test+&ie=utf-8&oe=utf-8&client=firefox-b-ab>
30. Rinott, R., Dankin, L., Alzate, C., Khapra, M.M., Aharoni, E., Slonim, N.: Show me your evidence – an automatic method for context dependent evidence detection. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, no. September, pp. 440–450 (2015). <https://doi.org/10.18653/v1/d15-1050>
31. Boltuzic, F., Šnajder, J.: Fill the gap! Analyzing implicit premises between claims from online debates. In: Proceedings of the 3rd Workshop on Argument Mining, no. August, pp. 124–133 (2016). <https://doi.org/10.18653/v1/w16-2815>
32. Habernal, I., Wachsmuth, H., Gurevych, I., Stein, B.: The argument reasoning comprehension task: identification and reconstruction of implicitwarrants. In: 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, NAACL HLT 2018, vol. 1, pp. 1930–1940 (2018). <https://doi.org/10.18653/v1/n18-1175>
33. Rubin, V.L., Lukoianova, T.: Truth and deception at the rhetorical structure level. J. Assoc. Inf. Sci. Technol. **66**(5), 905–917 (2015). <https://doi.org/10.1002/asi.23216>

34. Potthast, M., Kiesel, J., Reinartz, K., Bevendorff, J., Stein, B.: A stylometric inquiry into hyperpartisan and fake news. In: 56th Annual Meeting of the Association for Computational Linguistics Proceedings Conference, Long Paper, ACL 2018, vol. 1, pp. 231–240 (2018). <https://doi.org/10.18653/v1/p18-1022>
35. Shen, T., Lei, T., Barzilay, R., Jaakkola, T.: Style transfer from non-parallel text by cross-alignment. In: Advances in Neural Information Processing Systems 30 (NIPS 2017), vol. 30, no. Nips, pp. 6830–6841 (2017)
36. Hinton, G., Sabour, S., Frosst, N.: Matrix capsules with EM routing. In: International Conference on Learning Representations, ICLR, pp. 1–15 (2018). <https://doi.org/10.2514/1.562>
37. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: 31st Conference on Neural Information Processing Systems (NIPS 2017) Advances in Neural Information Processing Systems, vol. December, no. NIPS, pp. 3857–3867 (2017)
38. Jain, D.K., Jain, R., Upadhyay, Y., Kathuria, A., Lan, X.: Deep refinement: capsule network with attention mechanism-based system for text classification. *Neural Comput. Appl.* **32**(7), 1839–1856 (2019). <https://doi.org/10.1007/s00521-019-04620-z>
39. Li, S., Li, M., Xu, Y., Bao, Z., Fu, L., Zhu, Y.: Capsules based Chinese word segmentation for ancient Chinese medical books. *IEEE Access* **6**, 70874–70883 (2018). <https://doi.org/10.1109/ACCESS.2018.2881280>
40. Wu, Y., Li, J., Wu, J., Chang, J.: Siamese capsule networks with global and local features for text classification. *Neurocomputing* **390**, 88–98 (2020). <https://doi.org/10.1016/j.neucom.2020.01.064>
41. Du, Y., Zhao, X., He, M., Guo, W.: A novel capsule based hybrid neural network for sentiment classification. *IEEE Access* **7**, 39321–39328 (2019). <https://doi.org/10.1109/ACCESS.2019.2906398>
42. Kim, J., Jang, S., Park, E., Choi, S.: Text classification using capsules. *Neurocomputing*, **376**(2), 214–221 (2020). <https://doi.org/10.1016/j.neucom.2019.10.033>
43. Yin, H., Liu, P., Zhu, Z., Li, W., Wang, Q.: Capsule network with identifying transferable knowledge for cross-domain sentiment classification. *IEEE Access* **7**, 153171–153182 (2019). <https://doi.org/10.1109/ACCESS.2019.2948628>
44. Yang, M., Zhao, W., Chen, L., Qu, Q., Zhao, Z., Shen, Y.: Investigating the transferring capability of capsule networks for text classification. *Neural Netw.* **118**, 247–261 (2019). <https://doi.org/10.1016/j.neunet.2019.06.014>
45. Kumar, A., Narapareddy, V.T., Srikanth, V.A., Malapati, A., Neti, L.B.M.: Sarcasm detection using multi-head attention based bidirectional LSTM. *IEEE Access* **8**, 6388–6397 (2020). <https://doi.org/10.1109/ACCESS.2019.2963630>
46. Vlad, G.-A., Tanase, M.-A., Onose, C., Cercel, D.-C.: Sentence-level propaganda detection in news articles with transfer learning and BERT-BiLSTM-capsule model. In: Proceedings of the 2nd Workshop on NLP for Internet Freedom: Censorship, Disinformation, and Propaganda, no. November, pp. 148–154 (2019). <https://doi.org/10.18653/v1/d19-5022>
47. Gao, S., Ramanathan, A., Tourassi, G.: Hierarchical convolutional attention networks for text classification. In: Proceedings of the 3rd Workshop on Representation Learning for NLP, no. 2014, pp. 11–23 (2018). <https://doi.org/10.18653/v1/w18-3002>
48. Pennebaker, J.W., Booth, R.J., Boyd, R.L., Francis, M.E.: Linguistic inquiry and word count (LIWC). Mahw. Lawrence Erlbaum Assoc. **71**, 1–24 (2001). <https://doi.org/10.4018/978-1-60960-741-8.ch012>
49. Guu, K., Miller, J., Liang, P.: Traversing knowledge graphs in vector space. In: Proceedings of 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP, pp. 318–327 (2015). <https://doi.org/10.18653/v1/d15-1038>



Investigating Learning in Deep Neural Networks Using Layer-Wise Weight Change

Ayush Manish Agrawal^{1,5,6(✉)}, Atharva Tendle^{1,5,6(✉)},
Harshvardhan Sikka^{2,5,6}, Sahib Singh^{4,5,6}, and Amr Kayid^{3,5,6}

¹ University of Nebraska-Lincoln, Lincoln, USA

aagrawal@nebraska.edu, atharva.tendle@huskers.unl.edu

² Georgia Institute of Technology, Atlanta, USA

³ German University in Cairo, Cairo, Egypt

⁴ Ford R&A, Dearborn, MI, USA

⁵ Manifold Computing, 15805 Oakridge Road, Morgan Hill, CA 95037, USA

⁶ OpenMined, Oxford, Oxfordshire, UK

Abstract. Understanding the learning dynamics of deep neural networks is of significant interest to the research community as it can provide insights into the black box nature of neural nets. In this work, we conduct a study which analyzes layer-wise learning trends by measuring the relative change in the weights of a Deep Neural Net during training. Through our controlled yet exhaustive set of experiments we were able to identify key trends which could lead to better understanding of how neural networks learn and make way for better training regimes. In our work we explore the learning trends in ubiquitous convolutional neural networks and datasets. Our work provides a simple yet novel approach to interpreting neural networks and is different from previous investigative studies.

Keywords: Deep neural networks · Relative weight change · Convolutional neural networks · Learning trends

1 Introduction

Deep learning based approaches have achieved excellent performance in a variety of problem areas, generally consisting of neural network based models that learn mappings between task specific data and corresponding solutions. The success of these methods relies on their ability to learn multiple representations at different levels of abstraction, achieved through the composition of non-linear modules that transform incoming representations into new ones [1]. These transformation modules are referred to as layers of the neural network, and neural networks with several such layers are referred to as deep neural networks. Significant research has demonstrated the capacity for deep networks to learn increasingly complex functions, often through the use of the specific neural network primitives that

introduce information processing biases in the problem domain. For example, in the vision domain, Convolutional Neural Networks (CNNs) utilize convolution operations that use filtering to detect local conjunctions of features in images, which often have local values that are highly correlated and invariant to location in the image.

Various approaches have emerged that take advantage of the learning behavior of deep neural networks to improve their computational cost or reliability through interpretation. For example, transfer learning is a paradigm that focuses on transferring knowledge across domains, and often involves fine tuning neural networks that have been previously trained in a related domain to solve a new target task. This offers several advantages over training new networks from scratch on the task, as the prior learned parameters allow the network to learn the new task faster, assuming the pretraining domain is similar to the new one. Alongside this, a general observation in many computer vision tasks is that early layers converge to simple feature configurations [2]. This phenomena is observed in many vision architectures, including Inception and Residual Networks [3]. These findings, among others, point to a natural question: *Do different layers in neural networks converge to their learned features at different times in the training process?*

Understanding the layer-wise learning dynamics that allow for a deep neural network to learn the solution of a particular task is of significant interest, as it may provide insight into understanding potential areas of improvement for these algorithms and reduce their overall training costs. In this work, we empirically investigate the learning dynamics of different layers in various deep convolutional neural network architectures on several different vision tasks.

Our contributions are as follows:

- A metric to track the relative weight change in a given neural network layer on an epoch by epoch basis. We present relative weight change as a proxy for layer-wise learning, with the assumption that when the weights of a network have minimal change over a set of epochs, they are converging to their optimum.
- We track the relative weight change of several popular convolutional neural network architectures, including ResNets, VGG, and AlexNet for several benchmark datasets, including CIFAR-10, CIFAR-100, MNIST, and FMNIST.
- Learning dynamics are analyzed from the perspective of relative weight change for complex and simple learning tasks for shallow and deep networks with different architectural motifs. Several key trends emerge, including early layers exhibiting less relative weight change than later layers over the course of training across the CNN architectures.

The rest of this text is organized as follows: Sect. 2 presents related work. Section 3 introduces relative weight change and our experimental methodology. Section 4 discusses empirical results across several datasets and architectures. Finally, Sect. 5 discusses conclusions and future directions for this line of research.

2 Related Work

While explainability in deep learning is an active research area, most of the work in this field has been towards layer-wise feature visualization. Previous research has focused its efforts towards gaining a visual understanding of features as they convolve through a deep convolutional neural network that has provided the community with a visual understanding of what a network is learning [4–11]. In our work we aim to expand the boundaries of explainability but do so through an empirical approach involving the study of the layer weights. We differ from prior studies as our main contribution is a metric that computes the relative change in weights across epochs. This metric is then utilized to discover layer-wise learning trends during training. We investigate these trends through various architectures (Alexnet, VGG-19, ResNet-18) and datasets (MNIST, FMNIST, CIFAR-10, CIFAR-100) that are considered benchmarks in the deep learning research community. Through our experiments we document specific trends and discuss some higher level general trends that were discovered. We hope that our work provides researchers with a framework for improving end-to-end training regimes and helps them interpret how these networks learn.

3 Methods

3.1 Relative Weight Change

To better understand the layer-wise learning dynamics through the training process, we introduce a metric known as Relative Weight Change (RWC). RWC can be understood to represent the average of the absolute value of the percent change in the magnitude of a given layer’s weight. It can be formalized as

$$RWC_L = \frac{\|w_t - w_{t-1}\|_1}{\|w_{t-1}\|_1} \quad (1)$$

where L represents a single layer in a deep neural network, and w_t represents the vector of weights associated with L at a given training step t . We use the L_1 norm to characterize the difference in magnitude of the weights, and normalize the difference by dividing by the magnitude of the layer’s weights during the previous training step. Following this, an averaging step is applied to get a single value for RWC across the entire layer. The resulting proportion informs us as to how much the layer’s weights are changing over training steps. Smaller changes over a prolonged period indicate that the layer’s weights are nearing an optimum. We use this measure to characterize weight dynamics as on a per-layer basis as a function of training iterations to better understand how layers are learning.

3.2 Experimental Approach

Datasets and Settings. We use three benchmark datasets: CIFAR-10 [12] which contains 60,000 images of 10 classes, CIFAR-100 [12] which contains 60,000

images of 100 classes, MNIST Handwritten Digits [13], and FMNIST Fashion-MNIST [14] that contains 60,000 images of 10 classes. These benchmark architectures see significant use in deep learning research. The datasets also provide good variety in the complexity of their associated learning tasks. MNIST is fairly easy for simple networks to solve, FMNIST and CIFAR-10 provide new levels of complexity in image content and detail, and CIFAR-100 has significantly more classes and fewer samples per class, ramping up difficulty considerably.

Network Structure and Training. We use ResNet18 [15], VGG-19 with Batch Norm [16], and AlexNet [17]. These architectures were chosen for several reasons. They are ubiquitously used in the research community for computer vision problems. They also provide some variety in the information processing techniques and biases utilized to learn from images. For example, ResNets make use of residual connections, skip connections, and blockwise design while VGG makes use of significant downsampling and depth. A variety of architectures is useful for establishing some of the general trends we observe in this work, and inconsistencies may be attributable to the concrete differences between them. They also represent a good distribution of computational complexity, as AlexNet is significantly shallower than both ResNet and VGG variants.

The general training strategies used for these architectures was mostly consistent with those demonstrated in their respective papers. It's worth noting that the state of the art accuracy on our datasets required adaptive learning rate. We made a decision to exclude that from our training to focus on the layer-wise learning patterns in these deep networks. We used Stochastic Gradient Descent (SGD) [18] with momentum and weight decay for our experiments. The learning rate was kept constant throughout the experiments and each model was trained for a total of 150 epochs. Table 1, shows the detailed hyperparameters used for training on different architectures.

To interpret the layer-wise learning, we find the RWC as formulated in 1 for each layer per epoch. We run the same experiment for each architecture with different weight initializations using five different seeds to reduce the possibility of observing trends specific to a single run. We store the RWC array from each experiment, plot the average of the associated curves, and report the results in the following section.

Table 1. Detailed hyperparameters used for training

Architecture	Datasets	LR	Momentum	Weight decay
ResNet18	CIFAR-10	0.1	0.9	0.0001
	CIFAR-100	0.1	0.9	0.0001
	MNIST	0.1	0.9	0.0001
	FMNIST	0.1	0.9	0.0001
VGG19_bn	CIFAR-10	0.05	0.9	0.0005
	CIFAR-100	0.05	0.9	0.0005
	MNIST	0.05	0.9	0.0005
	FMNIST	0.05	0.9	0.0005
AlexNet	CIFAR-10	0.001	0.9	0.0001
	CIFAR-100	0.01	0.9	0.0001
	MNIST	0.1	0.9	0.0001
	FMNIST	0.1	0.9	0.0001

4 Results

Here, we include empirical results and analyses of layer-wise weight changes collected through the experimental approach described previously. Results are broken down by overall architecture, with trends highlighted for each of the four datasets. Figures demonstrating the RWC of specific layers are included and referenced in each set of analyses.

4.1 Residual Networks

The ResNet architecture is a deep convolutional network that consists of a repeated block motif of convolutional and batch normalization layers, along with residual connections between early and later layers. The convolutional hyperparameters of blocks are standardized. ResNet-18, used in these experiments, consists of four such residual blocks which we track explicitly as part of our analyses.

CIFAR-10 and CIFAR-100. Trained on CIFAR-10, ResNet-18 exhibits an increased relative weight change in later layers of the network as compared to earlier layers. Block 1 of the network, consisting of the first four convolutional layers following the input convolution layer, exhibits lower relative weight change over the duration of training as compared to Block 2. This can be seen in Fig. 1. Block 2 demonstrates a lower RWC as compared to Block 3, and Block 3’s relative weight change exhibits similar behavior to the last block of the network. These trends can be seen in Fig. 2 and Fig. 3, respectively. This instance of ResNet-18 achieved an accuracy of 91% on the test set provided by the PyTorch distribution of CIFAR-10. The similar scale of relative weight change in Blocks 3 and 4, coupled with the relatively good performance of the converged network

may indicate that ResNet-18 is able to learn the CIFAR-10 task without having to fully utilize the representational capacity of the last layers in the network present in Block 4. This interplay between complexity and the behavior of RWC in later layers of deep networks becomes evident in other results that follow. In general, we see that later layers demonstrate an increased RWC as compared to earlier layers in the network.

CIFAR-100 is a significantly more difficult task as compared to CIFAR-10, consisting of 100 classes for roughly the same number of data samples. Again, we see a trend of RWC increasing in later layers as compared to earlier layers through the course of training. Block 1 exhibits lower RWC as compared to Block 2, while Block 2 exhibits less RWC as compared to Block 3 in general. These trends can be observed in Fig. 4 and in Fig. 5. Interestingly, there is a noticeable difference between the RWC of Block 3 and 4, with the latter having a generally higher RWC (Fig. 6). This is in contrast to what was observed in CIFAR-10, where these blocks had similar RWC over the course of training. This difference may be the result of ResNet-18 using more of its representational capacity in later layers to solve the target task, as the CIFAR-100 task is significantly more difficult than CIFAR-10. The network achieved a 64% classification accuracy on the PyTorch distribution of CIFAR-100, further emphasizing the challenging nature of this particular classification problem and the increased relative weight change in later layers.

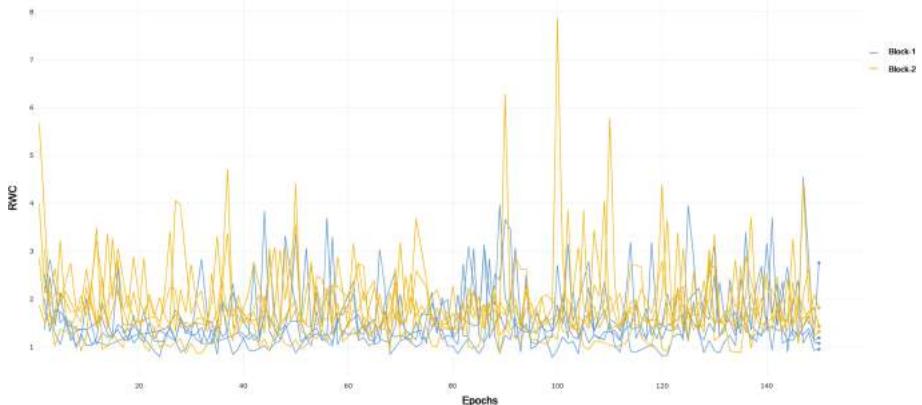


Fig. 1. RWC for ResNet-18 Blocks 1–2 on CIFAR-10

MNIST and FMNIST. Trained on MNIST, ResNet-18 exhibits a similar trend to CIFAR-10 and CIFAR-100, with RWC increasing in later Blocks as compared to earlier ones. Interestingly, we again see that Block 1 and Block 2 are lower than the later blocks and Block 4 exhibits lower relative weight change as compared to Block 3, mirroring the trend seen in CIFAR-10. We can observe this in Fig. 7, Fig. 8, and Fig. 9. This, along with the fact that ResNet-18 achieves a 99% test

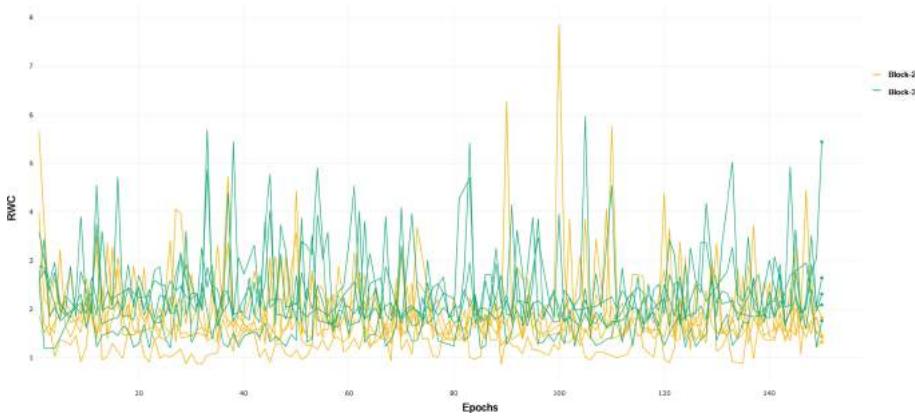


Fig. 2. RWC for ResNet-18 Blocks 2–3 on CIFAR-10

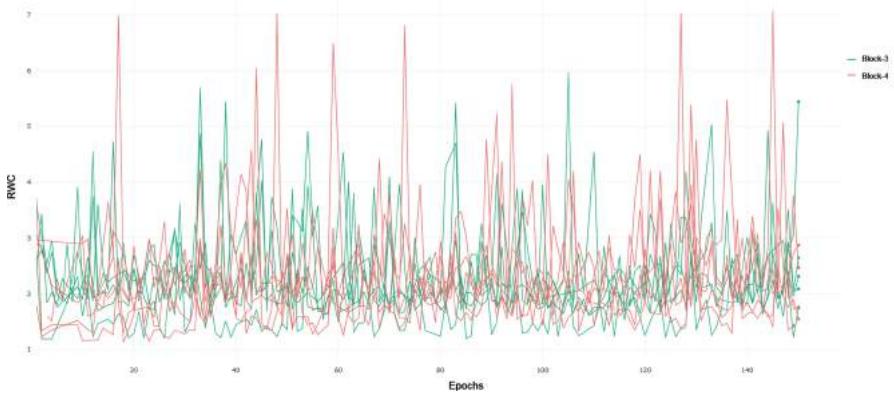


Fig. 3. RWC for ResNet-18 Blocks 3–4 on CIFAR-10

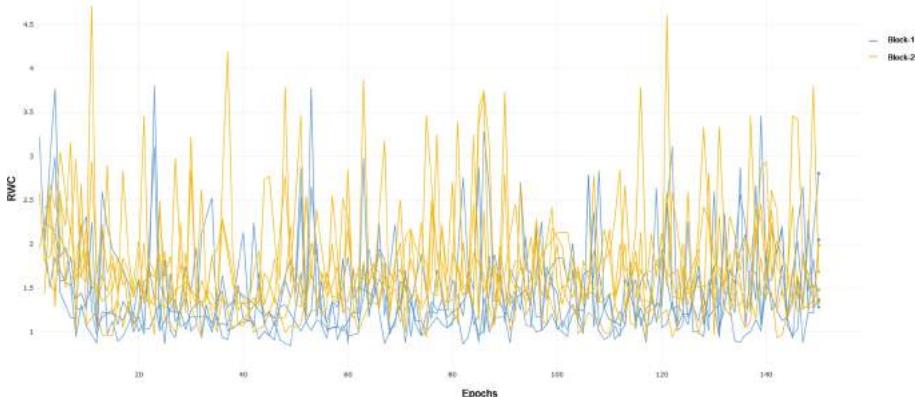


Fig. 4. RWC for ResNet-18 Blocks 1–2 on CIFAR-100

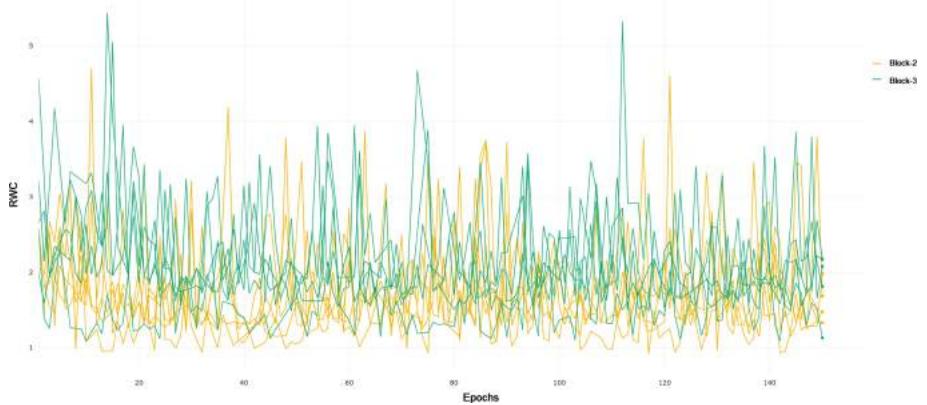


Fig. 5. RWC for ResNet-18 Blocks 2–3 on CIFAR-100

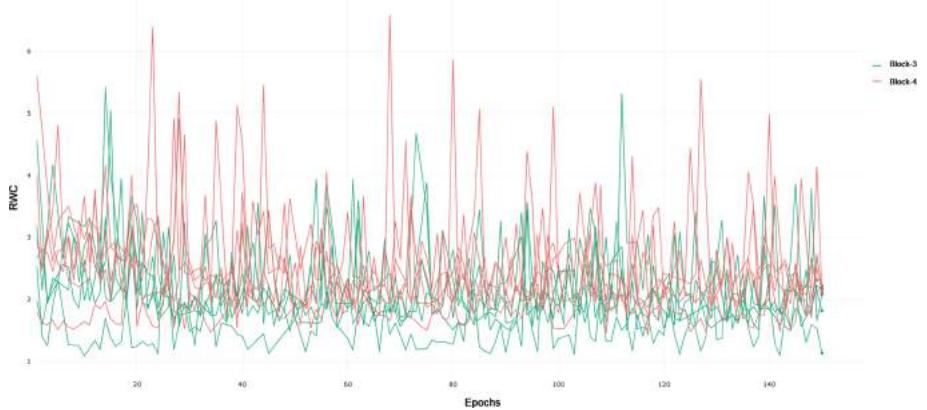


Fig. 6. RWC for ResNet-18 Blocks 3–4 on CIFAR-100

accuracy on MNIST, corroborates the interplay of complexity of the learning task and the capacity of the network, as MNIST is a simpler prediction problem and ResNet-18 likely does not need the full representational capacity of the layers in Block 4. FMNIST is a noticeably more difficult task with more complex images consisting of fashion objects rather than handwritten digits, and ResNet-18 achieves 92% performance on the test set. Blocks 1 and 2 exhibit lower RWC as compared to Block 3. Block 4 again is lower than block three, reflecting the same trend as seen in CIFAR-10 and MNIST, pointing to lower recruitment of the last few layers for the task. This can be observed in Fig. 10, Fig. 11, and Fig. 12. In general, the magnitude of RWC across all layers is observably lower for MNIST as compared to FMNIST, highlighting the increased difficulty and weight adjustments required to learn a solution for the latter task.

4.2 VGG

We use VGG-19 with batch norm due to its analogous depth when compared to ResNet-18. VGG-19 constitutes a more traditional convolutional neural network, stacking layers by down-sampling the images passed through the network. We compared the layer-wise learning by referring to the first four convolutional layers as earlier layers. Layers 5 through 11 were treated as middle layers, and layer 11 onwards were treated as later layers. We chose to divide the layers in this manner after noticing a common trend in the RWC in these layers as explained in the rest of this section.

CIFAR-10 and CIFAR-100. VGG-19 trained on CIFAR-10 exhibits similar behavior to ResNet-18 trained on CIFAR-10, where early layers exhibit lower relative weight change than middle and later layers. This can be observed in Fig. 13. Later layers again show lower relative weight change compared to the middle layers, demonstrating the same interplay of complexity and model capacity. For CIFAR-100, demonstrated in Fig. 14, we see that the general RWC is higher in magnitude across all layers. This trend may be due to the difficulty in the learning task. We again see middle and later layers with higher RWC compared to early layers, though the difference between middle and later layers themselves is less pronounced, pointing to more recruitment of later layers in a similar manner to what was observed in ResNet-18 on CIFAR-100. VGG-19 achieved a 90.5% and 63.6% test accuracy on CIFAR-10 and CIFAR-100, respectively.

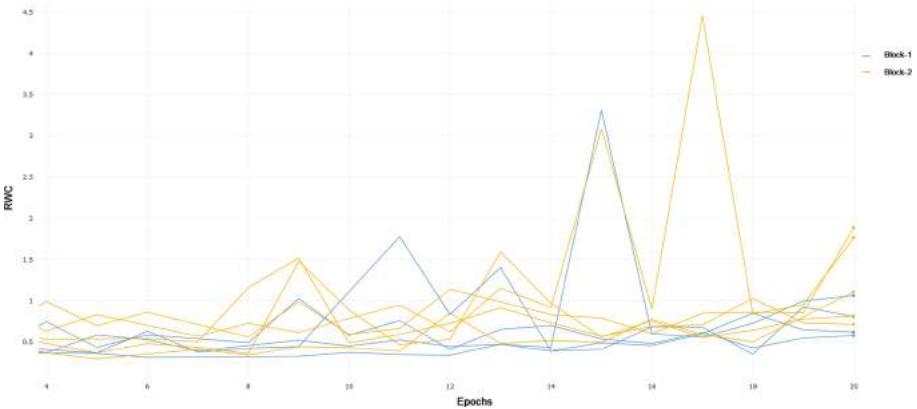


Fig. 7. RWC for ResNet-18 Blocks 1–2 on MNIST

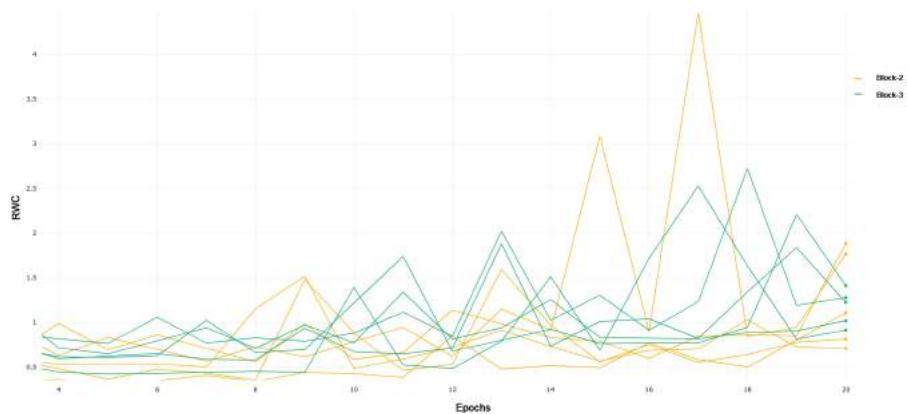


Fig. 8. RWC for ResNet-18 Blocks 2–3 on MNIST

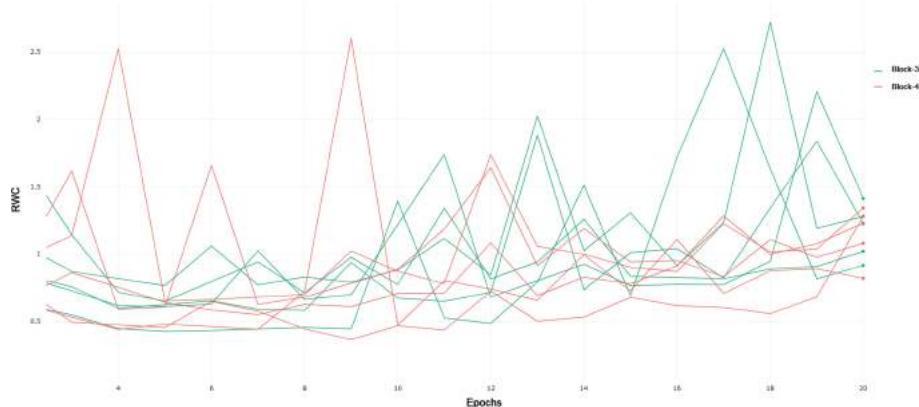


Fig. 9. RWC for ResNet-18 Blocks 3–4 on MNIST

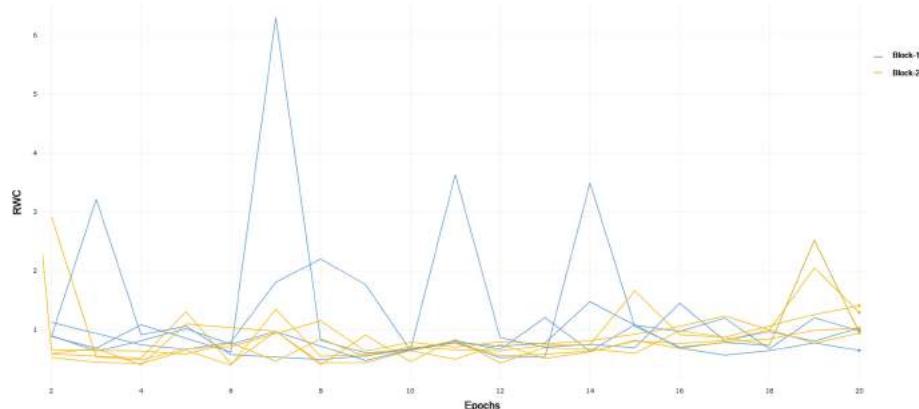


Fig. 10. RWC for ResNet-18 Blocks 1–2 on FMNIST

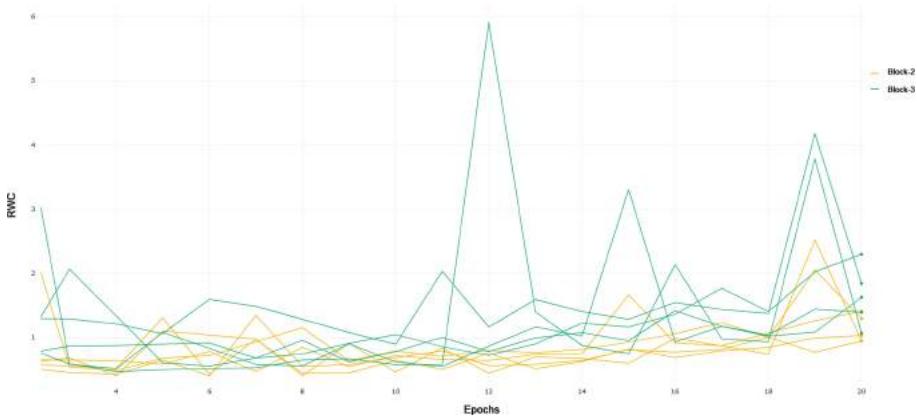


Fig. 11. RWC for ResNet-18 Blocks 2–3 on FMNIST

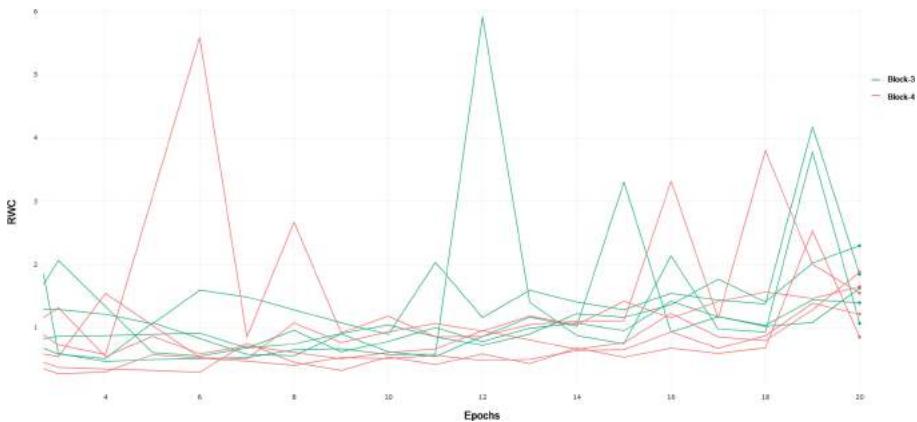


Fig. 12. RWC for ResNet-18 Blocks 3–4 on FMNIST

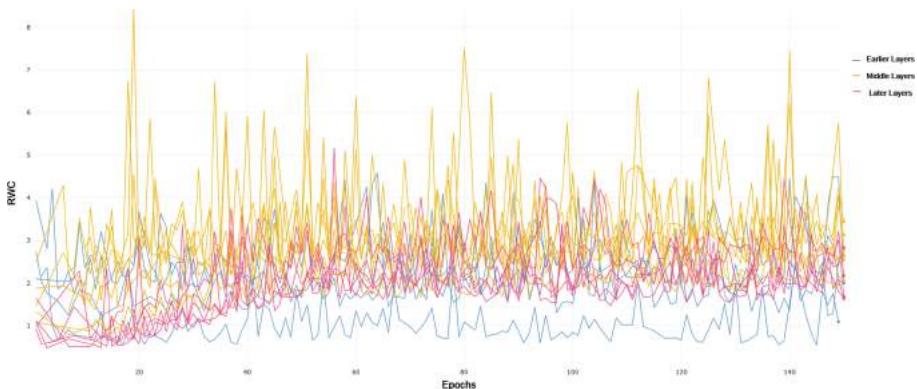


Fig. 13. RWC for VGG on CIFAR-10

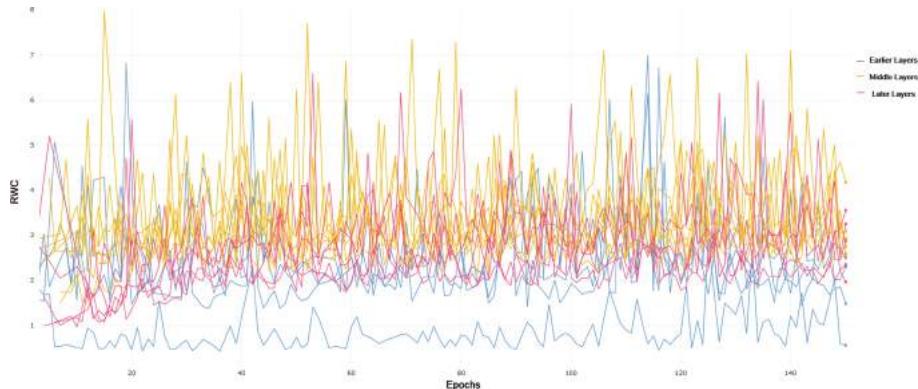


Fig. 14. RWC for VGG on CIFAR-100

MNIST and FMNIST. VGG-19 exhibits very similar trends in both MNIST and FMNIST, with early layers having lower RWC as compared to middle layers, but with later layers having the lowest overall RWC. This trend is evident in Fig. 15 and Fig. 16. Both of these learning tasks are relatively simple, and VGG-19 achieves 98.5% and 91.6% on MNIST and FMNIST, respectively. Lower overall RWC in later layers generally points to the same trend of deep architectures not needing to adjust learning in later layers as frequently for simpler tasks. Overall, the results and performance across both VGG-19 and ResNet-18 have a high degree of similarity. In general, the magnitude of RWC across all layers is again lower for MNIST as compared to FMNIST, similar to ResNet-18.

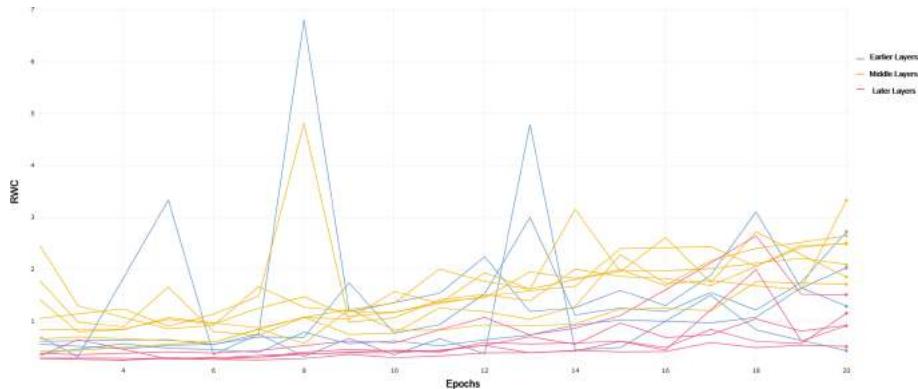


Fig. 15. RWC for VGG on MNIST

4.3 AlexNet

AlexNet is a simpler architecture as compared to VGG-19 and ResNet-18, and primarily serves as a benchmark to compare the layer-wise learning dynamics for a shallower convolutional network. The first convolutional layer is referred to as the early layer. The second and the third layers are the middle layers and the remaining two layers are referred to as later layers.

CIFAR-10 and CIFAR-100. Trained on CIFAR-10, AlexNet exhibits the same trends as the other networks, with early layers and later layers exhibiting lower overall RWC as compared to middle layers. This trend can be seen in Fig. 17. AlexNet demonstrates increasing RWC when trained on CIFAR-100, with early layer exhibiting less RWC as compared to middle layers and middle layers exhibiting less RWC as compared to later layers, which can be observed in Fig. 18. This is consistent with the behavior observed in ResNet-18 when trained on CIFAR-100, and may point to increased utilization of capacity in the network

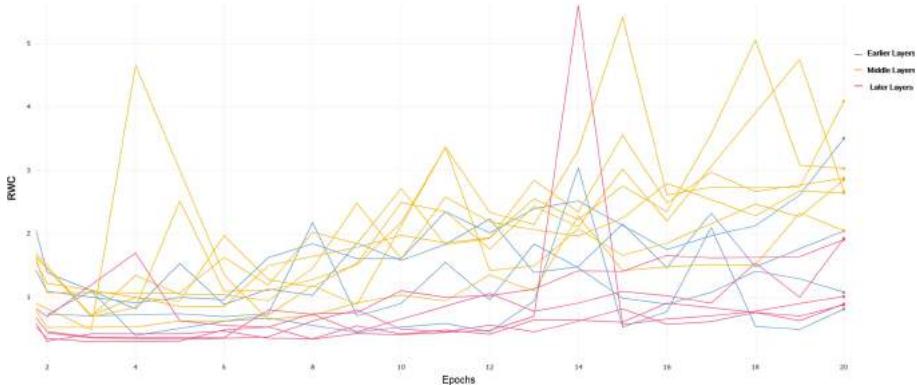


Fig. 16. RWC for VGG on FMNIST

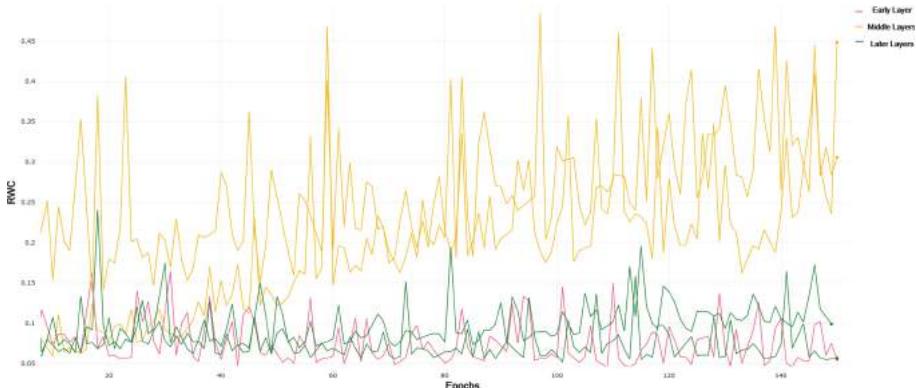


Fig. 17. RWC for Alexnet on CIFAR-10

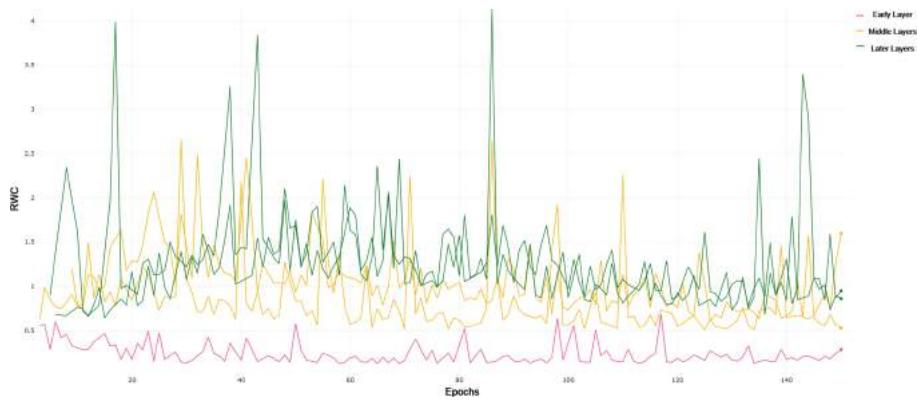


Fig. 18. RWC for Alexnet on CIFAR-100

for a much harder task. AlexNet achieved an 81% test accuracy on CIFAR-10, and a 53% test accuracy on CIFAR-100. These performances are to be expected, as AlexNet is a much shallower network.

MNIST and FMNIST. AlexNet trained on MNIST Fig. 19 and FMNIST Fig. 20 generally shows a trend of earlier layers having a lower RWC as compared to middle layers, and middle layers having a lower RWC as compared to later layers. In the same framework focused on the interplay between model capacity and task complexity, it seems that AlexNet uses its later layers' representational capacity for both MNIST and FMNIST. AlexNet achieves 98.5% on MNIST and 89.5% on FMNIST, respectively.

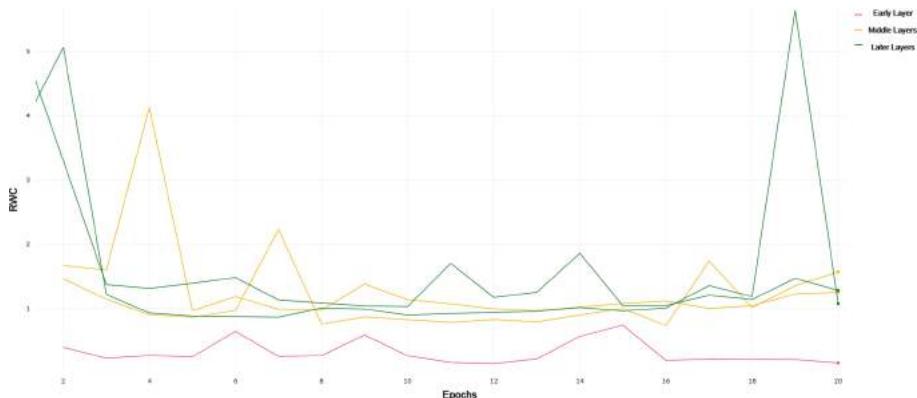


Fig. 19. RWC for Alexnet on MNIST

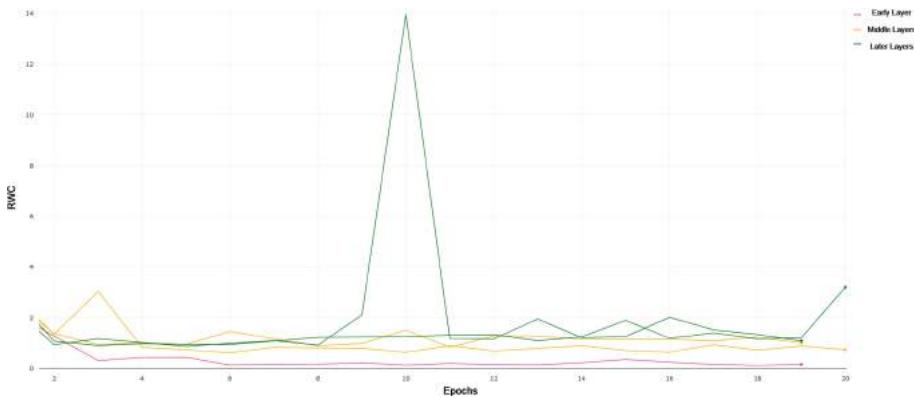


Fig. 20. RWC for Alexnet on FMNIST

5 Conclusion

In general, we see that relative weight change increases in later layers as compared to earlier ones across the different convolutional architectures, both deep and shallow, and across the different classification tasks. An interesting general trend emerges when networks are trained on comparatively simpler tasks like CIFAR-10 and MNIST, where later layers exhibit lower RWC as compared to middle layers of the network. On more complex tasks like CIFAR-100, we see that later layers exhibit higher RWC compared to early and middle layers, potentially indicating increased usage of the representational capacity of the network. Understanding layer-wise learning dynamics in deep networks provides a promising and impactful avenue of research, and has several potential future directions. These include the design of alternative metrics for layer-wise and neuron-wise learning in deep networks, pruning and freezing methods based on these metrics, and the empirical assessment of these metrics on other problem domains, including Natural Language Processing, Speech Recognition, and Reinforcement Learning.

References

1. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
2. Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In: Advances in Neural Information Processing Systems, pp. 3320–3328 (2014)
3. Cammarata, N., Carter, S., Goh, G., Olah, C., Petrov, M., Schubert, L.: Thread: circuits. Distill (2020). <https://distill.pub/2020/circuits>
4. Li, M., Zhao, Z., Scheidegger, C.: Visualizing neural networks with the grand tour. Distill (2020). <https://distill.pub/2020/grand-tour>
5. Erhan, D., Bengio, Y., Courville, A., Vincent, P.: Visualizing higher-layer features of a deep network (2009)

6. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: visualising image classification models and saliency maps (2014)
7. Nguyen, A., Yosinski, J., Clune, J.: Multifaceted feature visualization: uncovering the different types of features learned by each neuron in deep neural networks. arXiv preprint [arXiv:1602.03616](https://arxiv.org/abs/1602.03616) (2016)
8. Nguyen, A., Yosinski, J., Clune, J.: Understanding neural networks via feature visualization: a survey. In: Explainable AI: Interpreting, Explaining and Visualizing Deep Learning, pp. 55–76. Springer (2019)
9. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision, pp. 818–833. Springer (2014)
10. Olah, C., Mordvintsev, A., Schubert, L.: Feature visualization. Distill (2017). <https://distill.pub/2017/feature-visualization>
11. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2015
12. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
13. LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010)
14. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms (2017)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
16. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
17. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017)
18. Kiefer, J., Wolfowitz, J.: Stochastic estimation of the maximum of a regression function. Ann. Math. Stat. **23**, 462–466 (1952)



Deep Reinforcement Learning for Task Planning of Virtual Characters

Caio Souza and Luiz Velho^(✉)

Instituto Nacional de Matemática Pura e Aplicada, Estrada Dona Castorina 110,
Rio de Janeiro, RJ 22460-320, Brazil
lvelho@impa.br

Abstract. Intelligent agents are a long-standing area of interest in Artificial Intelligence. This paper aims to tackle the development of character controllers in an online hierarchical fashion. Unlike most previous works, we use deep reinforcement learning for the intermediate time scale planning and decouple the *motion synthesis* from *task planning* controller. While many challenges emerge from deep learning alone, we focus on fundamental reinforcement learning problems such as *temporal credit assignment* and *reward sparsity* and how the agent and environment modeling relates to this problem. Finally, we compare a dog controller using various designs (hand-crafted & visual input, continuous and discrete output encoding, reward and environment system modeling) in a fetch game scene, effectively building insight into task planning controllers development.

Keywords: Virtual characters · Intelligent agent · Task planning · Deep reinforcement learning

1 Introduction

Achieving a general intelligent agent is a long goal of AI. These agents have a broad range of applications, from automation to personal assistants and have been taking great advantage of recent advances of Deep Learning techniques [2, 11]. Current “state-of-the-art” agents are more focused on low-level controllers and working on specific narrow tasks or take advantage of task-specific context or protocols (i.e., [8, 14, 26, 29]).

Figure 1 presents a hierarchical abstraction of a broad agent. First, the low-level comprises the *motion synthesis*, which directly controls the agent actuators; next, the mid-level controls *task planning*: a series of actions directed by a goal; lastly, the high-level is in charge of *task selection* controlling the active task and goal. This type of abstraction bridges the concept of time-scale objectives in an explicit hierarchy.

Recently, there are extensive works in the literature covering motion synthesis [4, 6, 7, 9, 12, 15, 21–23, 31, 32, 34, 41–43], but works covering the mid and high

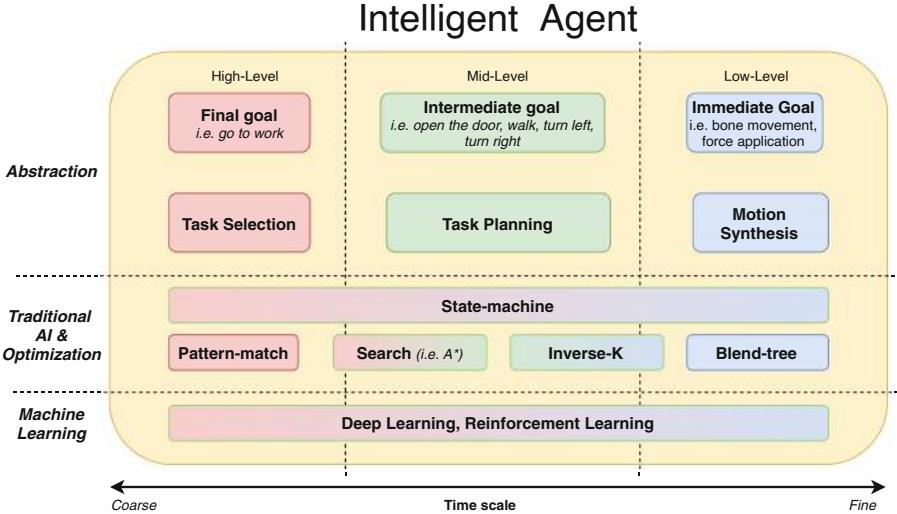


Fig. 1. Hierarchy levels of an intelligent agent and examples of solutions for each level.

levels are less numerous and usually are bound to specific low-level controllers or other specificities of complex fields such as robotics.

Here we investigate how to develop these task planning behaviors for virtual characters using deep reinforcement learning. The virtual character chosen is a dog named *DogBot*. For simplicity, it uses a traditional parametric low-level controller composed of state-machines, blend-trees, and standard animations, a mid-level controller comprising of the learned behavior and finally, the high-level is also a state-machine for trigger-based tasks. We evaluate various controllers for a fetching behavior using hand-crafted real-valued and third-person visual observations, continuous and discrete actions, and other environmental design choices.

Our modeling appeals to interactive storytelling applications where there is a guided narrative, but the non-playable characters need to adapt to dynamic environments. These applications impose some constraints, for example, the choice for the motion synthesis module, which usually requires naturalness more than physical accuracy for animations. Nevertheless, the explicit hierarchy and a guided narrative can also bring advantages. The first allows us to focus solely on the mid-level development and re-use the agent and environment for training with any motion synthesis module; the second will enable us to induce specific protocols (or expected actions) based on the narrative context simplifying the task selection. Lastly, deep reinforcement learning removes the computational burden of traditional planning methods based on optimization & search.

The rest of this manuscript is organized as follows: in Sect. 2, we link past and present works with our approach; next, in Sect. 3, we present a brief overview of our modeling and background of Markov Decision Processes. In Sect. 4, we dive

deeper into our modeling details. Our results for the learned controllers followed by a detailed discussion are in Sect. 5 (experiments). Finally, we finish with a few conclusions and directions for future works. Additionally, Appendix A and B contain other details that may be relevant for implementation and reproducibility.

2 Related Work

Developing intelligent agents tackling high to low-level controls (inclusive using vision-based sensing) are not new. Terzopoulos et al. [38] produced artificial fishes handling the motion synthesis controller with dimension reduction techniques and task planning with optimization, combining different sequences of low-level controllers. While it achieved good results, it required craftsmanship for each part of the system and had low scalability for today’s standards. Lin [18] investigated reinforcement learning usage with artificial neural networks for planning but was uncertain if it could scale for more difficult tasks. More recently, deep learning developments and its successful application with reinforcement learning achieving super-human performance on Atari games [27, 28] addressed the scalability issue.

Since the success with Atari games, Reinforcement Learning has been gaining momentum for motion synthesis varying from locomotion [9, 13, 19, 23, 26, 32, 33, 42]; locomotion with balance [20]; basketball dribbling [21]; body skill mimics such as cartwheel and backflips [31, 34]; drone flying [41] and others.

A few works use visual input sensing. Mnih et al. [27] uses a 2D view of Atari games for planning directed by a high-score goal. Levine et al. [16] uses an end-to-end scheme for training a 7-DoF robotic arm. While the input comes from a 3D environment, the camera position is fixed and resembles a view-from the top, being close to global sensing. Nakada et al. [30] uses a complex foveated 3D vision and accurate biomechanical physical motion synthesis for head-neck and limbs, but lacks full-body motion. Given its complexity, kinematic approaches addressing low-level controllers (i.e., [37, 43]) may be more suited for real-time applications expected to run on commodity hardware. Similar to ours, Merel et al. [25, 26] uses a first-person camera that translates to learning from a partially observable environment and needs additional sensing inputs for the task and proprioception, given its physics-based environment. Our approach relies on a third-person camera that stills partially observable and accounts for the kinematic character proprioception without the need for additional or hand-crafted inputs of the agent’s state.

Solving task-planning is relevant for many fields; classical solutions for planning involve search or optimization through state space, which can become intractable for high-dimensional or high-dynamic environments. In this regard, heuristics and online approaches have mainly been researched (and surveyed: heuristics for robotics [24], online [35]) trying to circumvent the shortcomings as mentioned earlier. Various techniques were developed in computer graphics, most of them tightly coupled to their motion synthesis modules. A few examples are:

using an A^* variation for dynamic path planning [17], applying optimization for foot-step planning [1], planning climbing routes also through optimization [29], using deep reinforcement learning for trajectory mimics [13].

Interestingly, the most recent work of Ling et al. [19] uses DRL to control locomotion tasks such as path following and maze runner. They call their approach a “model-then-control”, resembling our decoupled modules for motion synthesis and task planning, and uses a simple vision sensing with 16 rays for the maze runner task. We believe that decoupling each module allows for a better understanding and solution of each hierarchy level. For the specific case of task planning, focusing on the modeling of the learning environment work as a *learning methodology*, which can be applied to different motion synthesis modules, and hence can be a more general strategy.

3 Overview

Our agent implementation uses the *Unity3D* game engine and its machine learning framework *ML-Agents* [10]. A diagram of the agent modeling is shown in Fig. 2. The agent is split into three levels (Fig. 2 in yellow); first, the task selection communicating with the environment, waiting for specific conditions or triggers to select the appropriate behavior. Next, the task planner uses deep reinforcement learning; it takes various sensing of the environment and output actions such as turn left or right, synthesized into animations by the lower-level controller.

The realization of these schematics is our dog agent, later evaluated on a fetch game scene using different learned controllers. In the next section, we start with a brief background of Markov Decision Processes followed by detailed modeling of our agent and environment properties.

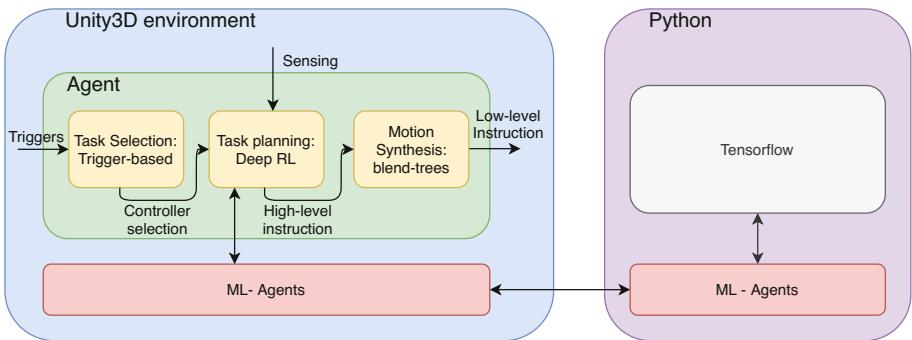


Fig. 2. Diagram of our agent modeling and the connections between each part. Note that the python connection is only used while training.

3.1 Background

Markov Decision Processes are the mathematical framework that idealizes sequential decision process. It is composed of a tuple (S, A, T, R) , where S is the set of states; A is the set of actions; $T : S \times A \times S \rightarrow [0, 1]$ is a probability distribution modeling the agent-environment dynamics with $T(s, a, s') = P(S_{t+1} = s' | S_t = s, A_t = a)$ and $R : S \times A \times S \rightarrow \mathbb{R}$ a function signaling the reward received when taking the action $a \in A$ at state $s \in S$ leading to the next state $s' \in S$.

Learning how to behave or a *policy* is the goal of MDP's. Deep reinforcement learning represents such policies as a parametric model, more specifically a neural network $\pi_\theta : S \times A \rightarrow [0, 1]$, which outputs the probability of choosing action a at state s . Here θ is the parameters, which can be learned by maximizing the expected reward:

$$J(\theta) = \mathbb{E}_{\tau \sim \zeta_\theta} [R(\tau)]$$

where τ is a trajectory $(s_0, a_0, r_0, \dots, s_n, a_n, r_n, s_{n+1})$ and $R(\tau) = \sum_{i=0}^n \gamma^i r_i$ is the total discounted reward of an experienced trajectory with discount factor $\gamma \in (0, 1]$. Methods using this formulation are known as policy gradient optimization.

The MDP abstraction fits very well with reinforcement learning as it covers a broad spectrum of problems. States and actions can be represented in various forms. The time steps t are not required to be equally spaced, and the intrinsic dynamics T and R are not needed to be known for learning, but trajectories or realizations of the process. Nevertheless, it is worth noting that the agent and environment design choices directly impact these dynamics, influencing the time complexity of the learning process and policy final's quality.

4 Environment and Agent Modeling

4.1 Environment

The environment is a Unity scene where the character can act and interact with objects. It consists of a square gray plane 110×110 m with a white border of 1 m diameter, which visually delimits the area of interest. Figure 3 shows the top view of the environment in the Unity editor.

The collectible in the agent's view is a pink box of 1 m side. Notice it is the collectible agent view, but a player in the scene can view any other complex geometry or rendering. It allows one to consistently reduce the scene complexity for the agent, which reduces the computational complexity of observations by reducing the visual observation's size, an inherent advantage of virtual environments.

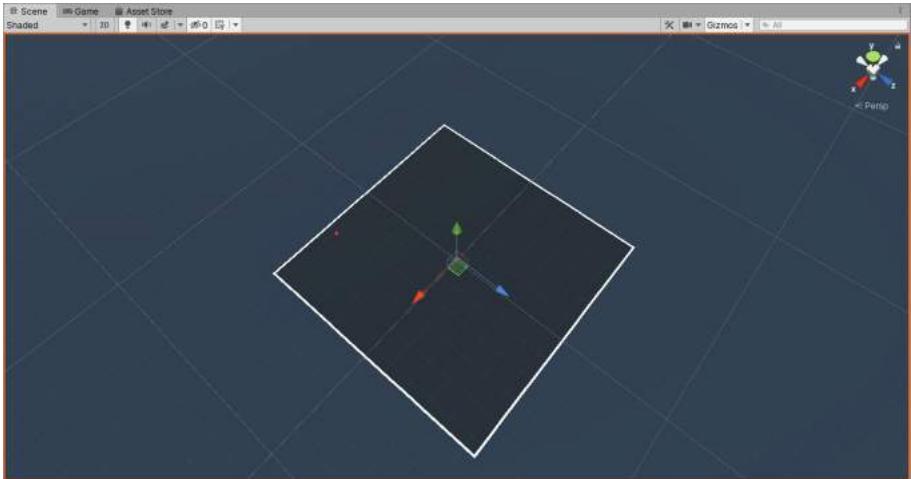


Fig. 3. Top view from the environment inside the unity editor.

Environment Observability: An essential conceptual distinction of environments is whether they are completely or partially observable. Although we describe as an environment property, observability is directly associated with the agent’s perception of its world. This differentiation of how much information the agent collects and how it perceives its surroundings is a significant learning performance component. Partially Observable Markov Decision Processes (POMDP) are usually harder to solve, so it must be accounted for when developing the agent’s observations.

4.2 Agent

The agent is an entity abstraction, which is itself a *behavior policy*. Nevertheless, we can imagine it as an entity with sensors gathering observations and actuators interacting where the policy link observations to actions. Our agent comprises all hierarchy levels, but the policy or controller learned is solely the task planning module. Notable features in an agent are: what it observes, how both the observations and actions are encoded, and how the associated rewards are assigned.

Agent Observation: An observation is any sensing made from the agent itself or the environment state encoded by numbers that serve as input to the behavior policy. Here, we employ two types of observations:

- Vector observation: composed of hand-crafted features:
 - Normalized direction to target: $d_{\text{target}} = (x, y, z)$, $\|d_{\text{target}}\|_2 = 1$
 - Normalized distance to border: $d_{\text{border}} = (x, y)$, $\|d_{\text{border}}\|_\infty < 1 \rightarrow \text{inside}$, $\|d_{\text{border}}\|_\infty \geq 1 \rightarrow \text{outside}$

- Linear velocity: $v_{\text{linear}} = (x, y, z) \text{m/s}$
 - Angular velocity: $v_{\text{angular}} = (x, y, z) \text{rad/s}$
 - Normalized agent forward direction: $d_{\text{forward}} = (x, y, z)$, $\|d_{\text{forward}}\|_2 = 1$
 - Normalized agent up direction: $d_{\text{up}} = (x, y, z)$, $\|d_{\text{up}}\|_2 = 1$
 - Agent local position (agent's position referent to the center of the environment): $p_{\text{local}} = (x, y, z)$
- Visual observation: 3rd-person-like camera aligned with agent forward direction down-sampled from the original rendered agent's view (Fig. 4):
- 2D image: matrix $I_{84 \times 84 \times 3}(r, g, b)$

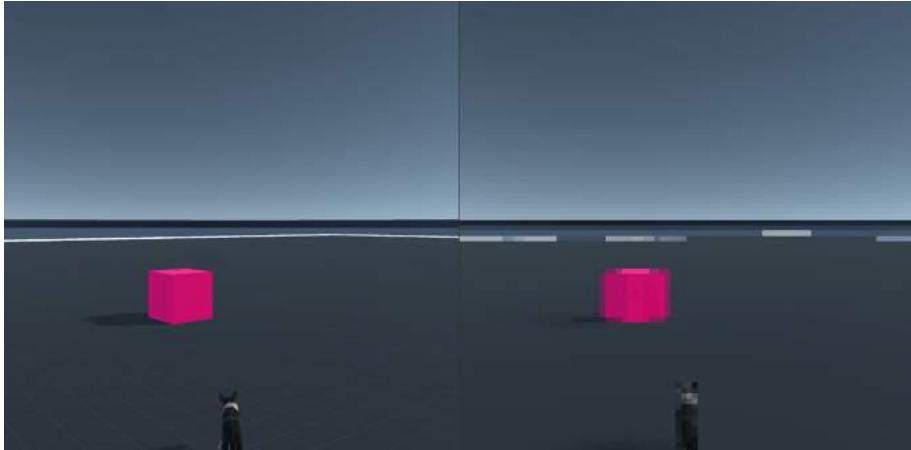


Fig. 4. Example of 3rd-person camera used for the agent. (Left) Agent's third-person camera. (Right) Downsampled visual observation.

Observation Constraint: We differentiate observations into two categories: Self-contained - Depends only on the agent state and sensors; Environment-dynamics dependent - Depends on the environment underlying mechanics. From those criteria, the 2D image is a self-contained agent sense. Concurrently, the presented vector observation needs access to the underlying environment dynamics to be fed to the agent (i.e., target position). The latter can limit the agent's usage in other unknown environments; however, the 3rd-person 2D image observations make the environment only partially observable.

Agent Actions: The DogBot's actions are defined by the motion synthesis module which is a character composed of various animations and a controller (state-machine and blend-tree) which receives four parameters controlling the X-Axis velocity, the Y-Axis rotation speed, and Boolean jump/crouch. Their encodings for continuous and discrete action space are:

- Continuous action space:
 - Forward and backward movement: $\in [-1, 1]$
 - Steering left and right: $\in [-1, 1]$
 - Jump: $j \in [-1, 1]$, *true* if $j > j_0$
 - Crouch: $c \in [-1, 1]$, *true* if $c > c_0$

where c_0 and j_0 are threshold parameters intrinsic to the agent controller.

- Discrete action space:
 - Forward and backward movement: {backward, none, walk, trot, run}
 - Steering: {left, none, right}
 - Jump: {true, false}
 - Crouch: {true, false}

Agent's Task: is a Fetch game - reach the collectible and bring it back. Its activation is trigger-based (i.e., throwing a stick) controlled by the task selection state-machine.

4.3 Reward

The entire universe of reinforcement learning bases itself on encouraging the best behavior through rewards (much like teaching a trick to a pet); in other words, rewarding the right actions accordingly. Developing a good reward signal is the key to learn an acceptable policy. Yet, most of the time, it is not easy to qualify a given action and state pair individually, but only the outcome of a sequence of actions and states. In theory, even for the cases where only the final state is rewarded, in the limit after many (infinite) experiences, it would be possible to learn an optimal behavior policy. As computational power and time are finite resources, engineering good reward systems are crucial. Here, our agent experiments with a sparse reward $-(+1.0)$ is given when the agent reaches its goal or final state, and per action reward $-r = +0.01(v_{\text{linear}} \cdot d_{\text{target}})$ which is positive when the agent moves toward its goal. For both cases, leaving the training area leads to a negative reward of -1.0 ending the episode. Also, a small negative reward -0.0005 is given at each time step t_i . It is close to $-\frac{1}{\# \text{ steps}}$, so the accumulated penalty will not saturate the total reward signal.

Note that the per action reward needs a broader knowledge of the environment, depending on both the agent and the environment's underlying state. Conversely, the sparse is assigned using only the agent's sensing but brings in the credit assignment problem. This notion is appealing because it resembles the agent's self-contained observation property, directly impacting the learning performance. In this case, even for the environment-dynamics dependent reward, it does not prevent the agent from being used in a new unknown environment after the learning process finishes, but for complex tasks may not be computable at all.

Next, we investigate the effects of the modeling choices and other variables, which are also part of the reward system, over the learned policies.

5 Experiments

For our experiments, we train various controllers from the combinations of environment and agent variations. For comparison purposes, we include another environment developed by Unity3D “Puppo, the Corgi” [39], which jointly learns the task planning and the low-level motion synthesis controller based on a physically simulated environment using joint torque and force.

5.1 Training

The tools used for training were Unity Editor 2019.3 and its machine learning framework ML-Agents [10] in its version v0.13.1 together with Tensorflow 1.14.0. All experiments use the Proximal Policy Optimization [36] algorithm and train for 10^7 steps. Experiments were run only once, given that they are computationally demanding. While the results are expected to be similar between different runs, but some variability should be taken into account when analyzing the results.

The training of the “Unity Puppo, the Corgi” uses their original configuration, besides the total number of steps. The environment is the same provided by [39], ported to the newer version of ML-Agents(v0.13.1) used here. The only modification to the original environment was its training area size to match DogBot’s area size.

Specific details of all parameters of our low-level controller, hyper-parameters, and network layout used for training are on Appendix A and B.

5.2 Results

Tables 1 and 2 contains the result of various trained controllers evaluated over 200 episodes (10^6 steps) on the test scene containing n collectibles which randomly re-spawn when collected. The evaluation metric is the *Score*, (the number of objects collected overall episodes) and *Reset*, (the number of times the agent was reset due to leaving the training area).

Figure 5 presents the comparison of Puppo and DogBot training convergence. This result is not directly comparable with the following Fig. 6 because, for comparison purposes, we use the same modeling from Puppo, which differs from ours in various ways. As expected, our approach to learning the task planning alone demands fewer steps. Our agent is also faster at completing the task; still, we cannot directly infer how efficient their policies are, given the differences in their low-level controller (i.e., maximum speed and minimum turning radius, etc.).

Table 1. Results for the various models on the standardized test scene. In order, the columns are: Experiment Number (Exp), Observation Type (Obs. Type), Action Type (Act. Type), Number of collectibles in the Training Environment (Train Env.), Reward Type (Reward Type), Number of collectibles in the Test Environment (Test Env.), Score (Score) and Reset (Reset).

Exp	Obs. type	Act. type	Train env.	Reward type	Test env.	Score	Reset
1	Vector	Discrete	1, box	Per action	1, box	1027	61
2	Vector	Continuous	1, box	Per action	1, box	2324	82
3	Vector	Discrete	1, box	Sparse	1, box	4	670
4	Vector	Continuous	1, box	Sparse	1, box	0	0
5	Visual	Discrete	1, box	Per action	1, box	1370	7
6	Visual	Continuous	1, box	Per action	1, box	2883	0
7	Visual	Continuous	24, boxes	Sparse	1, boxes	12	0
8	Visual	Continuous	24, boxes	Sparse	24, boxes	7119	3
9	Visual	Continuous	1, box	Per action	24, boxes	6163	0

Table 2. Results for the down-scaled environment on the standardized test scene. In order, the columns are: Experiment Number (Exp), Observation Type (Obs. Type), Action Type (Act. Type), Number of Collectibles in the Training Environment (Train Env.), Reward Type (Reward Type), Number of collectibles in the Test Environment (Test Env.), Score (Score) and Reset (Reset).

Exp	Obs. type	Act. type	Train env.	Reward type	Test env.	Score	Reset
10	Visual	Continuous	1, box	Per action	1, box	4606	599
11	Visual	Continuous	24, boxes	Sparse	1, box	1502	241

Figure 6 shows the progression of training from various configurations. The total reward is normalized to highlight the convergence of each controller. Interestingly, the controllers trained with sparse rewards vary from not learning anything (with a single collectible) to the faster showing progress in the learning (with multiple collectibles).

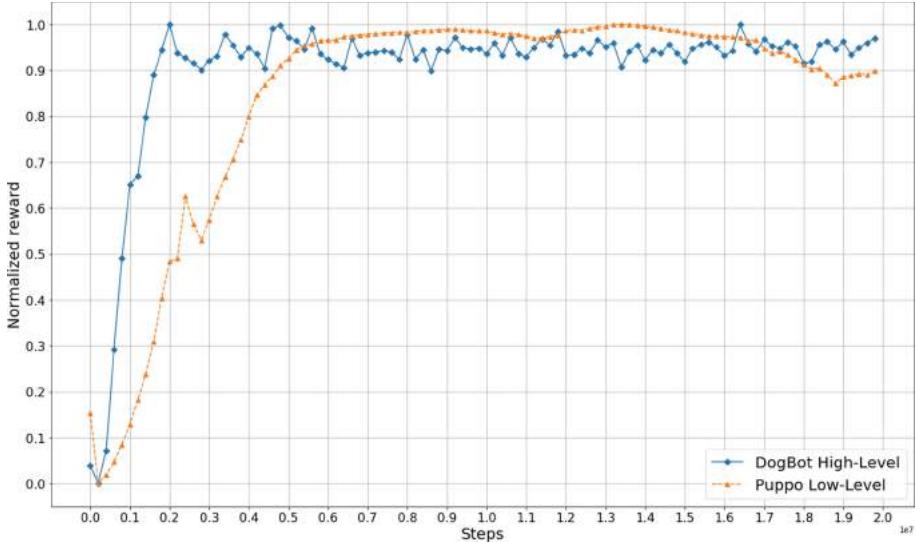


Fig. 5. Comparison of the training convergence of the “Puppo, The Corgi” which uses a low-level control approach (joint torque and force) versus the DogBot higher-level abstraction (Left, Right, etc.).

5.3 Discussion

This paper evaluated various mid-level controllers for task planning, which is a crucial role for intelligent agents. While the fetching task is simple, it is a good test-bed for accessing the impact of modeling choices in the agent’s final performance. The insight built from a simple case helps build more complex behaviors as the learning environment is de facto, where the developer has more control and understanding of its choices.

Our modeling coverage still far from extensive; purposefully, we left out network layout and hyper-parameters selection. While we agree these play a fundamental role in learning systems, they are much more experimental and sometimes hard to explain. In contrast, the concepts treated in our analysis for the fetch scene generalize more comfortably for other applications. In the following paragraphs, we individually analyze various aspects of modeling choices and their impact on the experiments.

Motion Synthesis and Task Planning Learning: Jointly training for low and mid-level control gives the opportunity of end-to-end learning (such as the Puppo), but in the present case, it converges slower than learning the task planning alone. Another downside is that it requires more effort for developing both the character and the learning agent, while the final animation result is not predictable for aspects such as naturalness.

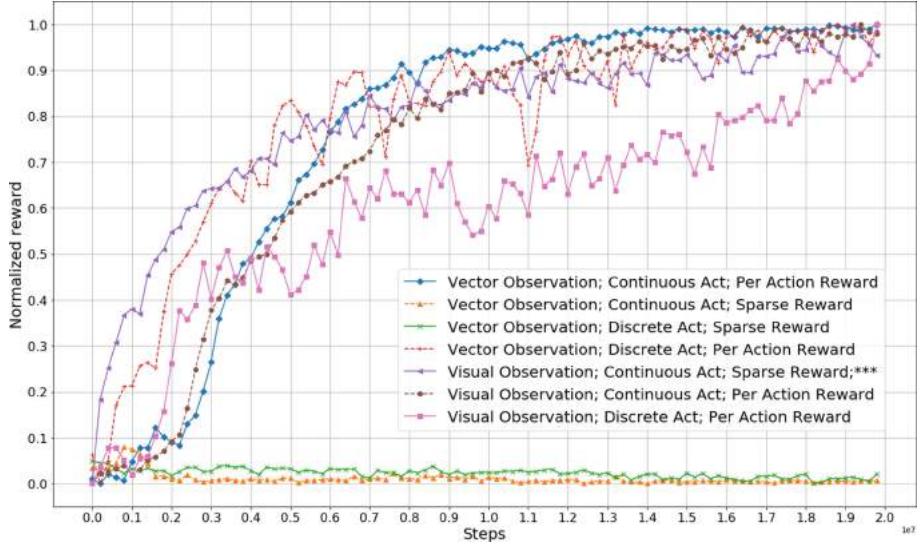


Fig. 6. Normalized reward progression during the training process. The naming scheme is the following: “*Observation Type*”; “*Action Type*”; “*Reward Type*”. The entry marked with *** Stands for the environment with multiple collectibles, while all others were using a single collectible.

Action Branches and Time Scale: One challenge when training with various action branches at the same time is the “curse of dimensionality” and the risk of not learning a proper policy. In our case, when starting from a random policy, the agent would be stuck alternating between going forward, backward, left, and right. A simple solution was to introduce a bias for the forward movement, but a more promising solution for this problem is bootstrapping the learning from examples. This approach showed to be effective in various works (i.e., [19, 26, 40]) and also has specific techniques for reinforcement learning, for example, Generative Adversarial Imitation Learning [5]. The time scale in which the agent can make decisions directly influences the learning; high frequency makes the agent more responsive but is harder for learning, starting from a random policy as the agent may not commit to a decision long enough to receive a reward.

Observation Type: Visual observations were as good or better than hand-crafted vector observations using the same reward system. Although they require more computational power, passing raw inputs applies to situations where hand-crafted observations are not viable. The only downside of raw inputs is the increased computational cost when training.

Action Space: Continuous action space achieved better results than the discrete action space, yet with slower convergence. We believe this result is directly related to the more precise control allowed by continuous actions, which requires more exploration explaining the slower convergence.

Reward: Using a per action reward system produced good outcomes in all cases, while sparse reward only worked in the scene with multiple objects. It is apparent that despite the agent's drawback of possibly exploiting the reward instead of learning the real objective, a per action approach is more reliable. Another crucial point is that the per action reward developed an acceptable exploration policy when the collectible was not visible. In these cases, the agent ran to the opposite side of the training area, while the agent trained with sparse reward, usually was stuck running in circles until it was close enough to notice the collectible. Having access to the underlying environment can help develop the reward system, but caution and extra work may be needed to achieve an appropriate per action reward. On the other hand, the sparse reward made the learning process entirely dependent on the environment's difficulty. A convenient approach in these cases could be using a curriculum [3], but requiring more human resources in the development process.

Multiplicity of Collectibles and Area Size: Having many objects in the scene completely changed the result of using sparse rewards, from unable to learn anything to achieve the best result in the test environment with multiple collectibles. This example shows how the environment modeling is crucial to learning; simple changes can make the initial random policy find rewards much often and learn faster. Interestingly, the agent trained with multiple collectibles underperformed with a single collectible test. From visual inspection, it did not develop an acceptable exploration policy and could not collect faraway objects. In contrast, in a reduced environment, the same agent performed better. Notwithstanding, while both agents could complete the task with a reduced area, they were not as effective as their original training area and would often leave the marked area. Indeed, training with such variation would lead to a better generalizing agent.

Limitations: Our present work has various limitations. The most pronounced of them is scaling relative to the number of controllers, which are linear on the number of behaviors and affects the complexity of implementing the higher-level selection of behaviors. Another critical point is the agent proprioception; while a third-person camera works well for our application, it may be inadequate for tasks requiring the agent's state's finer sensing. Lastly, while a hierarchical approach allows taking advantage of domain knowledge in each stage, not all applications can easily take advantage of prior knowledge or be split into well-defined control levels.

6 Conclusion and Future Work

We presented a hierarchical approach to developing intelligent virtual agents focusing on applying deep reinforcement learning to the intermediate time-scale level: task planning. Through a simple study case of a fetch behavior, we learned various controllers using different modelings and could build insight from each specific choice. This knowledge helps to make the base for future (more complex) works, from which we comment in the next paragraph.

A not extensive list of future directions is: Integrating more behaviors into our agent’s repertoire is unquestionably the next step; an exciting possibility for a dog agent are tricks such as hoop. Joining together these behaviors can lead to a fascinating virtual character for interactive narratives. Next, treating collisions and navigation in complex scenarios is a must for its applicability. Finally, more far fetched goals can be using more complex motion synthesis modules, integrating another sensing (i.e., audio) and memory to the agent. The memory of past events and actions is fundamental to specific long tasks; for example, a maze runner would be easier solvable if the agent keeps track of the visited spaces.

Conflict of Interest. The authors declare that they have no conflict of interest.

Appendix

A Motion Synthesis Module

The motion synthesis module is a Unity character controller responsible for receiving the task steps (i.e., left, right, jump, etc.) and translating to animations; it is the agent’s actuators. Its parameters control the agent’s velocity, turn speed, gravity, and other effects of the character’s physical properties. They are intrinsic to the agent’s actuators as they are not visible to the task planning module. In the Table 3, there is a detailed list of the values used for each parameter.

Table 3. DogBot’s character controller parameters.

Name	Value	Description
Moving turn speed	45.0 deg/s	Turn speed when not stationary
Stationary turn speed	30.0 deg/s	Turn speed when stationary
Jump power	5.0 m/s	Vertical velocity applied when jumping
Forward Velocity	9.0 m/s	Maximum forward velocity
Backward Velocity	2.0 m/s	Maximum backward velocity
Gravity multiplier	1.0	Multiplier for gravity simulation
Anim speed multiplier	1.0	Multiplier for animation time scale

B Training Configuration

In the Table 4, the training parameters are shown. The network architecture consists of the encoders and the decoders. The encoder for the vector observation consists of two dense layers fully connected with Swish activation function; for the visual observation the encoder is similar to [28] with three convolutional

layers of (# filters, kernel size, stride): (32, 8×8 , 4), (64, 4×4 , 2), (64, 3×3 , 1) and final dense layer with 512 units and leaky ReLU activation function instead of ReLU for of its all layers. Figure 7 presents a diagram of the visual encoder. The encoders' output is concatenated and fed to the decoders: the policy head and the value head. The first is responsible for choosing the agent actions, and its size varies according to its possibilities; the latter is liable for the state value estimate.

Table 4. Training parameters.

Name	Value	Description
Batch size (Continuous)	4096	Number of samples used for each optimization step for continuous action space
Batch size (Discrete)	256	Number of samples used for each optimization step for discrete action space
Buffer size	40960	Number of samples collected for each policy update
Hidden units	512	Number of neurons per hidden layer
Num layers	2	Number of hidden layers used for the model
Learning rate	3.0×10^{-4}	Initial learning rate for training
Max steps	2×10^7	Number of total simulation steps (actions) taken for training
Num epochs	5	Number of times each collected observation is used for training
Time horizon	1000	Horizon for learning, it represents how far in time steps one action can influence a past reward
Gamma	0.995	Discount factor, it represents how much of a n-future reward (R_n) is assigned to a present action in the form $\gamma^n R_n$
Curiosity strength	0.1	Strength of the curiosity intrinsic reward signal
Curiosity gamma	0.99	Discount factor for the curiosity reward
Visual encoding type	nature.cnn	Type of architecture used for the convolutional layers for the visual observation encoding
Visual input size	$84 \times 84 \times 3$	Size of the visual input image
Max episode length	5000	Maximum number of steps until an episode ends
Action repeat	2	Number of frames an action is repeated
Decision frequency	30 Hz	Frequency which the agent take decisions

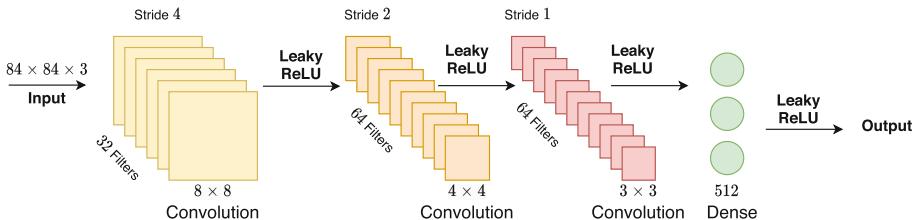


Fig. 7. Diagram of the network architecture used for the visual input encoder “Nature_cnn”.

References

1. Agrawal, S., van de Panne, M.: Task-based locomotion. *ACM Trans. Graph. (TOG)* **35**(4), 1–11 (2016)
2. Bengio, Y., LeCun, Y., et al.: Scaling learning algorithms towards AI. *Large-Scale Kernel Mach.* **34**(5), 1–41 (2007)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48 (2009)
4. Geijtenbeek, T., Van De Panne, M., Van Der Stappen, A.F.: Flexible muscle-based locomotion for bipedal creatures. *ACM Trans. Graph. (TOG)* **32**(6), 1–11 (2013)
5. Ho, J., Ermon, S.: Generative adversarial imitation learning. In: *Advances in Neural Information Processing Systems*, pp. 4565–4573 (2016)
6. Holden, D., Komura, T., Saito, J.: Phase-functioned neural networks for character control. *ACM Trans. Graph. (TOG)* **36**(4), 1–13 (2017)
7. Holden, D., Saito, J., Komura, T.: A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph. (TOG)* **35**(4), 1–11 (2016)
8. Hong, S., Han, D., Cho, K., Shin, J.S., Noh, J.: Physics-based full-body soccer motion control for dribbling and shooting. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
9. Jiang, Y., Van Wouwe, T., De Groote, F., Liu, C.K.: Synthesis of biologically realistic human motion using joint torque actuation. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
10. Juliani, A., et al.: Unity: a general platform for intelligent agents. arXiv preprint [arXiv:1809.02627](https://arxiv.org/abs/1809.02627) (2018)
11. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *Nature* **521**(7553), 436–444 (2015)
12. Lee, K., Lee, S., Lee, J.: Interactive character animation by learning multi-objective control. *ACM Trans. Graph. (TOG)* **37**(6), 1–10 (2018)
13. Lee, S., Park, M., Lee, K., Lee, J.: Scalable muscle-actuated human simulation and control. *ACM Trans. Graph. (TOG)* **38**(4), 1–13 (2019)
14. Lee, S., Ri, Yu., Park, J., Aanjaneya, M., Sifakis, E., Lee, J.: Dexterous manipulation and control with volumetric muscles. *ACM Trans. Graph. (TOG)* **37**(4), 1–13 (2018)
15. Lee, Y., Park, M.S., Kwon, T., Lee, J.: Locomotion control for many-muscle humanoids. *ACM Trans. Graph. (TOG)* **33**(6), 1–11 (2014)
16. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **17**(1), 1334–1373 (2016)

17. Levine, S., Lee, Y., Koltun, V., Popović, Z.: Space-time planning with parameterized locomotion controllers. ACM Trans. Graph. (TOG) **30**(3), 1–11 (2011)
18. Lin, L.-J.: Self-improving reactive agents based on reinforcement learning, planning and teaching. Mach. Learn. **8**(3–4), 293–321 (1992)
19. Ling, H.Y., Zinno, F., Cheng, G., Van De Panne, M.: Character controllers using motion VAEs. ACM Trans. Graph. (TOG) **39**(4), 40–1 (2020)
20. Liu, L., Hodgins, J.: Learning to schedule control fragments for physics-based characters using deep q-learning. ACM Trans. Graph. (TOG) **36**(3), 1–14 (2017)
21. Liu, L., Hodgins, J.: Learning basketball dribbling skills using trajectory optimization and deep reinforcement learning. ACM Trans. Graph. (TOG) **37**(4), 1–14 (2018)
22. Liu, L., Van De Panne, M., Yin, K.K.: Guided learning of control graphs for physics-based characters. ACM Trans. Graph. (TOG) **35**(3), 1–14 (2016)
23. Luo, Y.-S., Soeseno, J.H., Chen, T.-P.C., Chen, W.-C.: Carl: controllable agent with reinforcement learning for quadruped locomotion. arXiv preprint [arXiv:2005.03288](https://arxiv.org/abs/2005.03288) (2020)
24. Mac, T.T., Copot, C., Tran, D.T., De Keyser, R.: Heuristic approaches in robot path planning: a survey. Robot. Auton. Syst. **86**, 13–28 (2016)
25. Merel, J., et al.: Hierarchical visuomotor control of humanoids. arXiv preprint [arXiv:1811.09656](https://arxiv.org/abs/1811.09656) (2018)
26. Merel, J., et al.: Catch & carry: reusable neural controllers for vision-guided whole-body tasks. ACM Trans. Graph. (TOG) **39**(4), 39–1 (2020)
27. Mnih, V., et al.: Playing atari with deep reinforcement learning. arXiv preprint [arXiv:1312.5602](https://arxiv.org/abs/1312.5602) (2013)
28. Mnih, V., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529–533 (2015)
29. Naderi, K., Rajamäki, J., Hämäläinen, P.: Discovering and synthesizing humanoid climbing movements. ACM Trans. Graph. (TOG) **36**(4), 1–11 (2017)
30. Nakada, M., Zhou, T., Chen, H., Weiss, T., Terzopoulos, D.: Deep learning of biomimetic sensorimotor control for biomechanical human animation. ACM Trans. Graph. (TOG) **37**(4), 1–15 (2018)
31. Peng, X.B., Abbeel, P., Levine, S., van de Panne, M.: Deepmimic: example-guided deep reinforcement learning of physics-based character skills. ACM Trans. Graph. (TOG) **37**(4), 1–14 (2018)
32. Peng, X.B., Berseth, G., Van de Panne, M.: Terrain-adaptive locomotion skills using deep reinforcement learning. ACM Trans. Graph. (TOG) **35**(4), 1–12 (2016)
33. Peng, X.B., Berseth, G., Yin, K., Van De Panne, M.: Deeploco: dynamic locomotion skills using hierarchical deep reinforcement learning. ACM Trans. Graph. (TOG) **36**(4), 1–13 (2017)
34. Peng, X.B., Kanazawa, A., Malik, J., Abbeel, P., Levine, S.: SFV: reinforcement learning of physical skills from videos. ACM Trans. Graph. (TOG) **37**(6), 1–14 (2018)
35. Ross, S., Pineau, J., Paquet, S., Chaib-Draa, B.: Online planning algorithms for POMDPs. J. Artif. Intell. Res. **32**, 663–704 (2008)
36. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint [arXiv:1707.06347](https://arxiv.org/abs/1707.06347) (2017)
37. Starke, S., Zhao, Y., Komura, T., Zaman, K.: Local motion phases for learning multi-contact character movements. ACM Trans. Graph. (TOG) **39**(4), 54–1 (2020)
38. Terzopoulos, D., Rabie, T., Grzeszczuk, R.: Perception and learning in artificial animals. In: Proceedings of the 5th International Workshop on Artificial Life: Synthesis and Simulation of Living Systems (ALIFE-96), pp. 346–353 (1997)

39. Unity3D. Puppo, the corgi: Cuteness overload with the unity ML-agents toolkit (2018)
40. Vinyals, O., et al.: Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575**(7782), 350–354 (2019)
41. Jie, X., et al.: Learning to fly: computational controller design for hybrid UAVs with reinforcement learning. *ACM Trans. Graph. (TOG)* **38**(4), 1–12 (2019)
42. Yu, W., Turk, G., Liu, C.K.: Learning symmetric and low-energy locomotion. *ACM Trans. Graph. (TOG)* **37**(4), 1–12 (2018)
43. Zhang, H., Starke, S., Komura, T., Saito, J.: Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph. (TOG)* **37**(4), 1–11 (2018)



Accelerating Deep Convolutional Neural on GPGPU

Dominik Żurek^(✉), Marcin Pietroń, and Kazimierz Wiatr

AGH University of Science and Technology,
al. Adama Mickiewicza 30, 30-059 Krakow, Poland
{dzurek,wiatr}@agh.edu.pl

Abstract. This paper is focused on the improvement of the efficiency of the sparse convolutional neural networks (CNNs) layers on graphic processing units (GPU). The Nvidia deep neural network (cuDnn) library provides the most effective implementation of deep learning (DL) algorithms for GPUs. GPUs are one of the most efficient and commonly used accelerators for deep learning computations. The modern CNN models need megabytes of coefficients and needed millions MAC operations to perform convolution. One of the most common techniques for compressing CNN models is weight pruning. There are two main types of pruning: structural (based on removing whole weight channels) and non-structural (removing individual weights). The first enables much easier acceleration, but with this type it is difficult to achieve a sparsity level and accuracy as high as that obtained with the second type. Non-structural pruning with retraining can generate a matrix-weight up to /90% or more of sparsity in some deep CNN models. This work shows when is worth using a direct sparse operation to speed-up the calculation of the convolution layers. The VGG-16, CNN-non-static and 1×1 layers from ResNet models were used as a benchmarks. In addition, we present the impact of using reduced precision on time efficiency.

Keywords: CNN · GPU · Pruning · cuDnn · Cublas · Reduced precision

1 Introduction

Deep convolutional neural networks (CNNs) achieve outstanding result in various artificial intelligence tasks including image classification [9, 12], object detection [18], semantic segmentation and natural language processing [8, 22, 23]. Recent CNN neutral networks consist of dozens of cases of convolution and a few fully connected layers. Neural networks for conducting the training process on large benchmark datasets need different accelerators such as multi-core processors, GPGPUs or other dedicated hardware accelerators. Over the years, scientists have been looking for methods to accelerate the calculations of the convolution operation. The direct convolution algorithm to perform convolutions requires N^2 multiplications and $N(N-1)$ additions where N is the size of the input. For the

same input the fast Fourier transform (FFT) method reduces operation complexity to $O(N \log_2(N))$ [1]. The Winograd algorithm is suitable for small fixed-size kernels and requires 2.25 times fewer multiplications than direct convolution [11]. The convolution operation can be realized by matrix-multiplication [4], especially on the GPGPU which is highly tuned for performing this operation [5]. The GPGPU remains one of the most efficient and commonly used hardware accelerators. The NVIDIA deep neural network library (cuDNN)¹ depends on filter size, batch size and data representation; it performs convolution with different algorithms (Winograd, fft, gemm). By contrast some CNN models for image processing or natural language processing can be heavily pruned. The effect of this process is that they very often contain zero values more than 80% of weights. Depending upon the level of sparsity, it can be worth performing the convolution through the application of the direct sparse convolution method proposed by *Chen* [3]. This paper is focused on investigating when it is worth using sparse operations, instead of using dedicated NVIDIA libraries to perform the convolution layer on the GPGPU. As the main optimization strategy we propose the introduction of a unified sparse level for each of the output channels in each convolutional layer. The other optimization strategy is determining the most optimal number of thread blocks for each convolutional layer separately. The presented approach is optimized towards the optimal arrangement of the data in order to obtain acceleration with the direct convolution approach using the sparse format. These strategies are crucial for achieving peak performance. Finally, the impact of using the half precision (FP-16) in direct sparse convolution on time efficiency is explored. It is compared with cuDnn, where for 16-bit data representation, NVIDIA Tensor Cores specialised arithmetic units are used. To our current knowledge, this is the first work that shows the acceleration of the unstructured sparsity of weights compared to the dense approach using real models.

In any application in which deep learning models are applied, the speed of the calculation plays the major role. This particular aspect is limited in the GPGPU by the efficiency of the algorithms which are provided by the cuDnn library. The main goal of our research is to find the solutions which are able to perform the convolution operation quicker than the dedicated library. In this paper, the speed increase is gained by reducing the number of MAC operations during the calculation of the convolution, which is possible by taking advantage of the fact that multiplying by zero does not affect the final result so could be omitted.

The paper is organised as follows. In Sect. 2, works related to the subject matter are presented. Section 3 presents some theoretical background relating to the convolution neural networks. Section 4 presents methodology for the calculation of the convolution operation on graphic cards. Section 5 describes the process of building the CSR weight format. Section 6 demonstrates the conducting of experiments. Section 7 summarises the obtained results. Section 8 closes the paper and provides conclusions.

¹ <https://developer.nvidia.com/cudnn>.

2 Related Work

Convolution complexity and efficiency optimization have recently become quite popular research subject. Jordá *et al.* [7] presents how the way in which the cuDnn library calculates convolution layers depends upon parameter configurations and data representation. Lavin *et al.* [11] introduces Winograd convolution implementation which is based on minimal filtering algorithm. This approach for small filter and batch size was 2.26 times faster than the contemporary of cuDnn. Adámek *et al.* [1] proposes FFT based convolution on GPGPU by shared memory implementation of the overlap-and-save method, and for certain sizes, a 30% speed increase was achieved in comparison to cuDnn. The direct sparse convolutions method was proposed in [13]. The authors used the CSR format to store the weights and perform the convolution operation by use of the sparse matrix multiplication. This approach achieved 3.1–7.3 times speed increase comparison to dense convolution in the AlexNet model, on Intel Atom, Xeon and Xeon Phi processors. Lu *et al.* [14] proposed FPGA’s sparse convolution implementation which in VGG16 is almost three times faster than FPGA’s dense implementation. The same type of convolution was applied on GPU [3], where the speed increase for AlexNet [10], GoogleLeNet [20] and ResNet [6] models were respectively 1.74, 1.34 and 1.43 times in comparison to GEMM implementation (see Sect. 4.1) in the CUBLAS² library. Zhu *et al.* [24] used sparse matrix operation in order to perform recurrent neural networks (RNN), where the data format of sparse persistent RNN are represented by the $\langle index, value \rangle$ pairs. The authors have proposed a several optimization strategies for GPU implementation such as wide memory loads and bank-aware weight layouts. This approach for a hidden layer of size 1782 and density of 10% allows the following speed increases to be achieved: 7.3 times compared to dense GEMM (cuDnn), 3.4 times compared to sparse GEMM (cuSparse³) and 1.8 times compared to dense persistent implementation (cuDnn).

3 Convolutional Neural Networks

The typical convolutional layer in a feedforward procedure calculates the convolution of the inputs which is represented by a batch of N samples (images, time series etc.) with C channels and size $H \times W$, with the set of K filters with C channels and size $R \times S$. The output product of convolution contains K matrices with size $E \times F$, where $E = \frac{H+2\circ padding-R}{stride} + 1$ and $F = \frac{W+2\circ padding-S}{stride} + 1$. The set of parameters of a single convolution layer is a 4D array called a *tensor*. When kernel is marked as W and the input is marked as I then the convolution of a single layer is given by the formula:

$$Out_{n,i,j,k} = \sum_{c=0}^{C-1} \sum_{r=0}^{R-1} \sum_{s=0}^{S-1} W_{k,c,r,s} I_{n,c,i+r,j+s} \quad (1)$$

² <https://developer.nvidia.com/cUBLAS>.

³ <https://docs.nvidia.com/cuda/cusparse>.

The result of the above formula is added to the bias parameter b and the activation function is then applied [17]. The last layers of the most popular CNN model usually are fully connected layers that take the output of the last CNN layer, turn it into a single vector and apply weights to compute the class score. The convolution layers are the most time consuming operation in the CNN flow. For this reason, only these layers are taken for consideration in this paper.

3.1 VGG-16 Model

The VGG-16 is a convolutional neural network model [19], which won the ILSVR (Imagenet) competition in 2014⁴. VGG-16 has each convolution layer of 3×3 filters with a stride 1 and each pool layer uses a max function of 2×2 filter with a stride 2. At the end, there are three fc layers followed by a softmax function. The input has a fixed size $224 \times 224 \times 3$ (RGB format). This is very large network in which the number of parameters is approximately 138 million. The placement of the layers is shown in Fig. 1. In this paper, we attempt to improve each vgg-16's convolution layer as an example of 2D convolution.

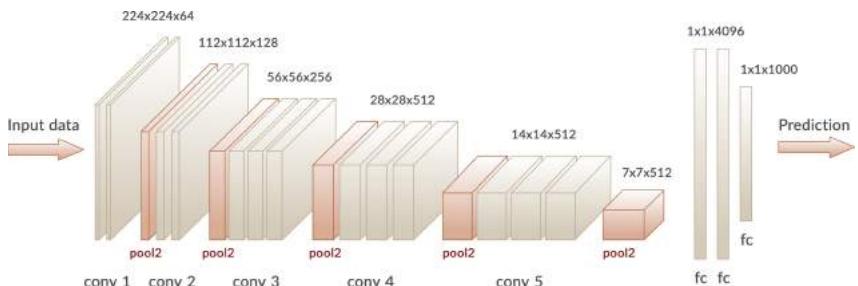


Fig. 1. VGG-16 architecture

3.2 CNN-non Static

CNN-non-static [8] is commonly used natural language processing(NLP) model for sentence classification tasks. The model contains two CNN layers where both contain 128 filters with size of 2×300 and 3×300 respectively. The results of these layers are connected to dense layers with 128 outputs followed by a softmax function. As an input 64×300 pre-trained vectors are used, which are produced by GloVe embeddings [16]. In our experiments, this model is used as an example of 1D convolution to examine the possibility of improving this kind of convolution through usage of sparse representation.

⁴ <http://image-net.org/challenges/LSVRC/2014/>.

3.3 1×1 Convolutions

A 1×1 (or pointwise) convolution is mainly used to change the depth between the input and output volumes, which is very useful for decreasing the number of dimensions before expensive convolution such as 3×3 or 5×5 . In ResNet [6] architecture that kind of layer reduces number of dimensions for 3×3 convolution and after this restores the original shape. This feature is presented in Fig. 2.

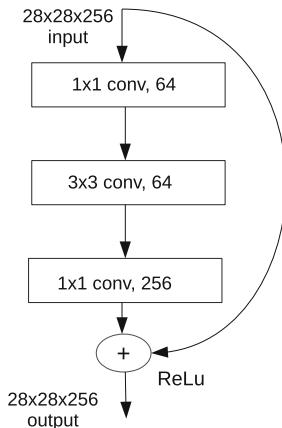


Fig. 2. Usage of 1×1 convolution in ResNet model

4 Convolution Algorithms on GPGPU

In order to perform forward convolution the cuDnn library always chooses the most effective algorithm, depending upon input, filter and batch size. It is possible to enforce which algorithms should be run, but it is not an effective strategy. In this paper's simulation, cuDnn has been used to decide which method should be applied to perform the convolutionnp. The graphics processing units (GPUs) are very effective particularly for accelerating large matrix products such as matrix-matrix multiplication and element-wise multiplication. For this reason, the most productive algorithms for performing convolution on GPGPUs first of all transform the data to the form which allows performing the convolution through the application of these operations.

4.1 General Matrix Multiply (GEMM)

This method transforms the input and the filters into two matrices. The matrix I with the $CRS \times EF$ dimensions is created from the input data, by duplicating

the original data in order to overlap the filter position in the convolution. As a result, the input tensor contains $R \times S$ more data, which requires more memory. The second matrix W with a size of $K \times CRS$ is generated by reshaping filters. Convolution is performed In such way the by the scalar product of the single row and the single column which is repeated for each input's column and all rows from transposed filters. As a result, the output O matrix is created with a size of $K \times EF$. As it turns out, this method is used in 1D and 1×1 convolution and when the number of channels is relatively small, which usually takes place for the first layers of most cnn's architecture.

4.2 Fast Furrier Transformation

The fast Furrier transformation (FFT) can be used to calculate convolution because the Fourier transform of the convolution of two signals in one domain (e.g. time) is equivalent to the point-wise multiplication of their Fourier transform in the other domain (e.g. the frequent domain). If \mathcal{F} denotes the Fourier transform, \parallel denotes convolution and \bullet denotes point-wise multiplication, then the convolution of the two functions f and g can expressed as:

$$f \parallel g = \mathcal{F}^{-1}(\mathcal{F}(f) \cdot \mathcal{F}(g)) \quad (2)$$

The inverse fast Fourier transform \mathcal{F}^{-1} is requested in order to return back to the time domain. The FFT convolution is more effective on account of the fact that on the GPGPU, calculation of the dot product is faster than the calculation of convolution even if there is a requirement for two transformations. Based on VGG-16 architecture, the cuDnn uses this method to perform convolution in the case of the input size being smaller than or equal to 58×58 (from the sixth layer) and for a batch size higher than 32. In theory, FFT convolution is the most effective way to perform convolution for large filters. For these conditions, the cuDnn chooses the FFT_TILING algorithm, which means that the input is divided into tiles with a fixed size.

4.3 Winograd

The Winograd convolution algorithm [11] is suitable for small filter sizes (denoted as $m \times m$) such as 3×3 . In such cases the input image of size $p \times p$ where p is bigger than 4, is divided into 4×4 overlap with stride 2 in order to perform convolution. Each created tile and filter is transformed through multiplying by special matrices B and G , respectively, to a form which allows performing convolution by element-wise multiplication. Finally, the result of this is transformed by multiplication through special matrix A , into a 2×2 matrix which is the result of convolution with the size $(p - 2) \times (p - 2)$. The special matrices B , G and A contains only -1 , 1 and 0 values and their construction allows producing a minimal filtering algorithm base on the Chinese remainder theory (CRT) [21]. Thanks to these transformations there is a reduction in the number of multiplications with regard to $\frac{p^2}{m^2(p-2)^2}$, and an immediate increase in the number of

required additions, which results in faster computation. If D denotes the input matrix, F is the filter matrix and \bullet is element-wise multiplication, the result S is calculated by the formula:

$$S = A^T \sum [GFG^T] \bullet [B^T DB] \leftarrow \quad (3)$$

5 Building CSR Weight Format

After the training process incorporated with incremental pruning, the model contains a set of weights with values set to zero. From the point of view of this paper, the most important element of pruning is the extracting information the lowest sparsity level occurs with the K output channels. This information is used to unify the sparsity level for each output channel. This procedure is significant for GPU implementation, where the execution time depends upon the output channel with the lowest sparsity (see Sect. 6). Having a standardized sparsity level, pruned weights enables the compressed sparse row (CSR) format for each convolution layer to be built. To represent the matrix, the CSR format needs to build three arrays:

- *Values* - these contain only non-zero elements.
- *Coldix* - on each position contains a offset for the value on equivalent position in the *value* array. Park et al. [15], proposed pre-computed value in the coldix matrix to store indexes from the input array which will be used to perform convolution. Thanks to this, there is no necessity to calculate these indexes during convolution which decrease calculation time.
- *Rowptr* - $rowptr[i]$ is the point to the first non-zero element of the i th output channel. Note that the result of $rowptr[i+1] - rowptr[i]$ is the number of non-zero elements in the i th output channel. In our approach, this number is the same for each output channel thanks to the aforementioned standardized sparse level and we call this the *sparse_level*. The only modification which must be done to avoid calculating the sparsity level separately for each output channel is mark some zero value as “non-zero” and during building, the CSR format treats them as normal value. This operation does not change the result and needs extra-memory. However, having the same sparsity for each output channel determines that each warp on the GPGPU has the same number of iterations, which is known before running the CUDA kernel which leads to the kernel’s faster execution. These special “non-zero” values are chosen to not excessively jump thorough memory this mean there are choose zeros with side-by-side indices to guarantee contiguous direct memory access.

6 Convolution Operation Using a Sparse Operation on GPGPU

To perform convolution by usage of a sparse operation *the direct sparse convolution* [15] was used. In parallel implementation we use an approach proposed

by [3]. The input data are stored in NCHW format (batch size, channel, height, width). The weights are stored in the CSR format where *coldix* and *value* arrays are loaded into shared memory. Each single thread block, calculates one output channel so for one input vector the total number of thread blocks is equal to the number of output channels. In our approach, we optimize the number of input vectors from the input batch, which will be handled by this number of blocks and is denoted as *subBatchSize*. As a result, the total number of thread blocks is equal to $\frac{\text{batchSize} \times \text{number of Output Channel}}{\text{subBatchSize}}$ which is presented by Fig. 3.

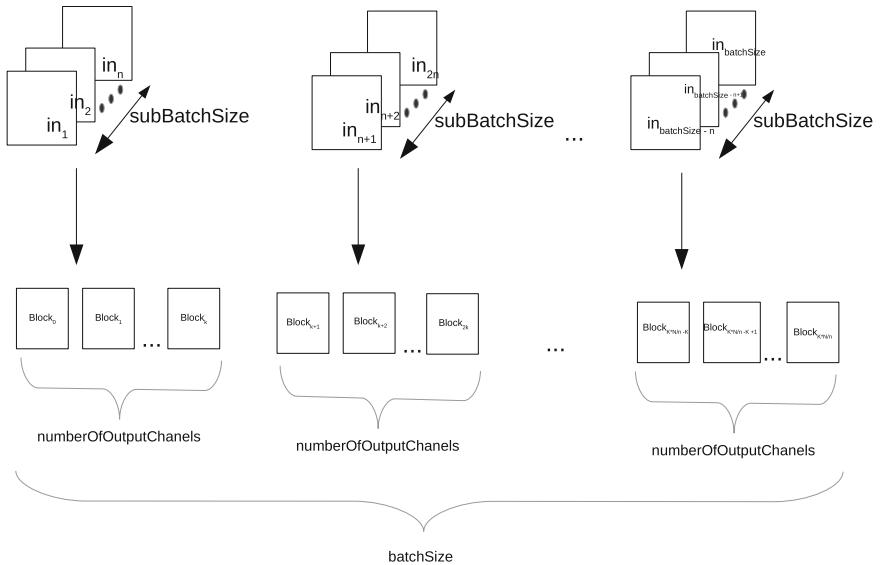


Fig. 3. The total number of thread's block using to perform convolution for a single layer

As it turns out, this number depends on the layer size and belongs to $\{2, 4, 8\}$ (see Sect. 7). This optimal number is not the same for each type of layer due to the cache limitation and when this particular memory is missed, data are put into global memory which is very slow. For this reason, in this paper this number was experimentally fixed for each layer. During the calculation of the convolution, non-zero values from the *values* array and pre-calculated indices of the input vector from the *coldix* array are loaded from shared memory into the thread local memory and it is reused for *subBatchSize* input vectors. This procedure enables maximum limitation of the reading from shared memory. Similar to the weights and indexes values, the partial sums are stored in registers and are copied to global memory after calculations. The number of threads used for the calculation of one output channel for one vector is determined by the output size of convolution. Each single thread is responsible for calculating one single output value by multiplication with the weight with corresponding input

value, accumulating the partial sum and writing the final result to the global memory as is shown in Fig. 4. The total number of working threads in a single threads block is determined by the sparsity level which in our approach is the same for each output channel and is equal to sparsity of channel with minimum value (see Sect. 5). In the version of the implementation where each channel has different sparsity the execution time was longest $\geq 28\%$ for VGG-16 3x3 and 1x1 convolution type and $\geq 26\%$ in the case of both convolution layers from CNN-non static. As an improvements both weights and input feature maps are marked as constant in order to hold them in the L2 cache, and coalesced memory access is provided. This convolution function is chosen for the cuDnn flow and performs convolution instead of the cuDnn function in the case of specific sparsity level (higher than $\geq 90\%$ for vgg-16 and 1x1 convolution, and more than $\geq 78\%$ for CNN-non static) and this is achieved only for some layers, as is shown in the next sections. The greatest acceleration of direct sparse convolution over the cuDnn was achieved for the 1D convolution. In this case, the input data are in the shape of a vector; therefore to preform convolution by usage of the *direct sparse method*, less memory jumps are needed than with 2D convolution. Besides sparsity we are checking how precision reduction can accelerate the calculation of convolution in sparse implementation and with usage of dedicated libraries. In order to achieve this, data are transformed from *float* to *half* type for both weights (the *value* array) and input data. Cuda-Math-Api⁵ is used in order to perform calculations with half the precision on the GPU. Cuda-Math-Api provides transformations and mathematical functions for *half* type. As described in [2] and [6] the 16-bit half precision is sufficient to keep the CNN models accuracy on the same level.

Table 1. Time results for VGG-16

Convolution size (CHWK)	Escoin time [ms] - float	cuDnn time [ms] - float\algorithm	Escoin time [ms] - half	cuDnn time [ms] - half\algorithm
3x224x224x64	2.48	2.82\GEMM	2.21	2.71\GEMM
64x224x224x64	60.73	19.07\WINOGRAD	27.08	31.82\GEMM
64x112x112x128	16.45	10.56\WINOGRAD	8.87	9.48\GEMM
128x112x112x128	28.11	17.28\WINOGRAD	17.31	15.88\GEMM
128x56x56x256	13.61	9.21\FFT-TILING	8.74	7.81\GEMM
256x56x56x256	23.72	14.27\FFT-TILING	15.83	16.09\GEMM
256x28x28x512	9.34	6.72\FFT-TILING	6.07	7.86\GEMM
512x28x28x512	16.06	15.02\FFT-TILING	14.01	16.82\GEMM
512x14x14x512	4.31	4.84\FFT-TILING	4.18	4.66\GEMM

7 Results

All of the presented calculations were performed on the Nvidia Tesla V100-SXM2-32GB⁶. The batch size is always equal 128 (for others or 64 or 256, the

⁵ <https://docs.nvidia.com/cuda/cuda-math-api/>.

⁶ <https://www.nvidia.com/en-us/data-center/v100/>.

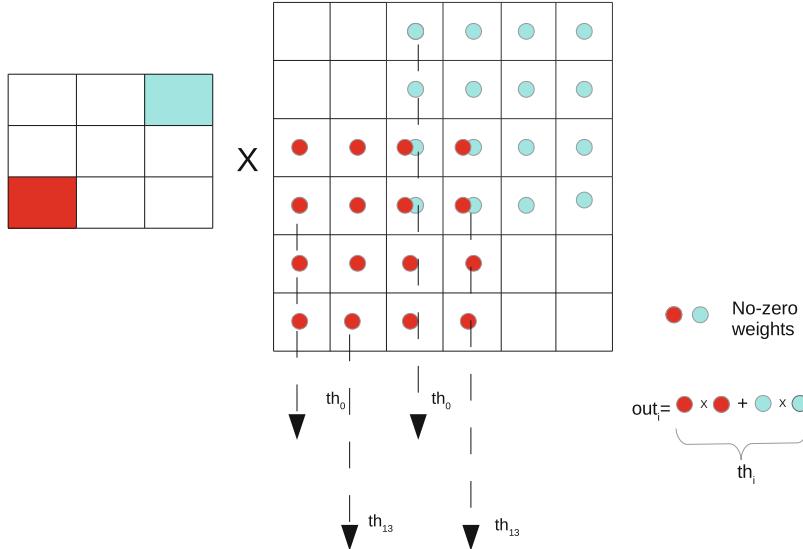


Fig. 4. Calculating convolution using a sparse operation

Table 2. Time results for ResNet. 256 filters $1 \times 1 \times 64$

Data type	CUDNN time\algorithm	Escoin time
Float	0.35\GEMM	0.32
Half	0.30\GEMM	0.27

proportions are the same) and the final execution time is calculated as the average of 10 iterations. In our experiments, we let the cuDnn library choose the algorithm which would be used to perform convolution for different layers and data types. The tables below, presents the algorithm which was used by the cuDnn library in addition to the execution time for each layer. For each experimental calculation of convolution the *subBatchSize* was determined (see Sect. 6) which for last three layers from VGG-16 and for both 1×1 convolution layers from ResNet50 is 8. For the remaining VGG-16 layers and for CNN-non static with filter size two, this value is 4, and for CNN-non static with filter size three, this parameter is 2. The presented results were measured with the optimal value of this parameter. Without setting this value by the method proposed in this paper,

Table 3. Time results for ResNet. 64 filters $1 \times 1 \times 256$

Data type	CUDNN time\algorithm	Escoin time
Float	0.37\GEMM	0.31
Half	0.29\GEMM	0.24

it would not be possible to achieve better performance than cuDnn because when the number of block is equal to $numberOfOutputChannel \parallel batchSize$, for VGG-16 and 1×1 layers, the performance decreased by around $\geq 10\%$. In the case of CNN-non static, the decrease was $\geq 12\%$. An even larger drop in performance occurred when all the data from the batch was processed by K blocks this value was between $\geq 38\%$ and $\geq 45\%$. Table 1 includes the results of time execution for the VGG-16 model (see Sect. 3.1) model, where for each layers the sparsity was set at $\geq 90\%$ because this is the lowest sparsity level for which the *direct sparse convolution* algorithm is more effective than the cuDnn library. Despite such a high sparse level, the improvement over the cuDnn was not achieved for every layer. Only for the first and last three layers where the input size in NCHW format, is $3 \times 226 \times 226$ and $512 \times 16 \times 16$ respectively, was the improvement gained for *float* ($\geq 13\%$ -first layer and $\geq 12\%$ -last layer) and *half* ($\geq 22\%$ -first layer and $\geq 11\%$) data type. In addition, in both cases the algorithm is faster for the *half* type which is not obvious for the cuDnn library, where for *half-precision*, the cuDnn always performs convolution by the *GEMM* algorithm (see Sect. 4.1). This way of calculating the convolution on *half* type, for the VGG-16's convolution layers with input sizes $64 \times 226 \times 226$, $256 \times 58 \times 58$ and $512 \times 30 \times 30$ is less effective than performing this on *float* type with the use of *FFT* (see Sect. 4.2) or *WINOGRAD* (see Sect. 4.3) algorithm. Taking into account only the data in *half* type format, the sparse approach can additionally improve performance of the VGG-16's conv layers with the follow input size: $64 \times 114 \times 114$, $128 \times 58 \times 58$, $256 \times 30 \times 30$, $512 \times 30 \times 30$. Having the same level of sparsity as in the VGG-16 architecture, there is the possibility to achieve better performance than the cuDnn for the 1×1 convolution in Resnet architecture (see Sect. 3.3). For this type of convolution, the cuDnn always uses the *GEMM* algorithm and the result for this are included in Tables 2 and 3. The most effective performance of the *direct sparse convolution* method is achieved for the 1D convolution which is dedicated to the time series data. A significant acceleration compared to cuDnn was reached for the CNN-non static (see Sect. 3.2), where for convolution layer with kernel size 2, the sufficient sparsity level is $\geq 77\%$ to gain a $\geq 9\%$ and $\geq 11\%$ speed increase for the *float* and *half* data types, respectively (see Tables 4 and 5).

Table 4. Time results for CNN-non-static for input 300×64 . Kernel size 2

Data type	CUDNN time\algorithm	Escoin time for given sparsity		
		77%	83%	87,5%
Float	0.192\GEMM	0.176	0.126	0.102
Half	0.161\GEMM	0.145	0.097	0.069

Table 5. Time results for CNN-non-static for input 300×64 . Kernel size 3

Data type	CUDNN time\algorithm	Escoin time for given sparsity		
		77%	83%	87,5%
Float	0.231\GEMM	0.236	0.188	0.135
Half	0.204\GEMM	0.182	0.148	0.103

8 Conclusions

This work is focused on speeding up the convolution operation on GPGPU through the use of the sparse matrix operation and the representation of data at a reduced level of precision. In particular, this strategy makes maximum use of knowledge about the number of produced zero values as a result of the pruning process. The time results obtained from the proposed solution are comparable with the convolution kernel from the cuDnn library, which is recognized as the most effective way to perform convolution on GPGPUs. We have presented concrete cases when it is worth performing convolution using the *direct sparse convolution* in the cuDnn place. The most improvements are archived for 1D convolution because for this type, the cuDnn library always chooses the GEMM method to perform convolution which does not provide such a strong performance such as the WINOGRAD or the FFT-TILING method which are used to performing 2D convolution. It is shown that 2D convolution using *direct sparse convolution* can also outperform cuDnn algorithms. The efficiency of the proposed solution is highly dependent on the level of sparsity. For this purpose, for future work we will consider devising and implementing a pruning method which allows us to achieve a high level of sparsity and improve the accuracy of large DL models, which is possible when they are used on less complex and reduced datasets. Additionally, we have examined the influence of conducting the calculation using reduced precision on time efficiency. These experiments show that using low precision of data during computation can significantly speed up the calculations and save memory. This fact is an incentive to devise a quantisation method which enables conducting convolution operations through the use of 8-bit without a noticeable drop in accuracy.

Acknowledgment. This work has been supported by the funds provided by AGH University of Science and Technology in 2020.

References

1. Adámek, K., Dimoudi, S., Giles, M., Armour, W.: GPU fast convolution via the overlap-and-save method in shared memory (2019)
2. Al-Hami, M., Pietron, M., Casas, R., Wielgosz, M.: Methodologies of compressing a stable performance convolutional neural networks in image classification, January 2020
3. Chen, X.: Escoin: efficient sparse convolutional neural network inference on GPUs (2018)

4. Chetlur, S., et al.: cuDNN: efficient primitives for deep learning (2014)
5. Dongarra, J.J., Hammarling, S., Higham, N.J., Relton, S.D., Valero-Lara, P., Zounon, M.: The design and performance of batched BLAS on modern high-performance computing systems. In: ICCS (2017)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
7. Jordà, M., Valero-Lara, P., Peña, A.J.: Performance evaluation of cuDNN convolution algorithms on NVIDIA Volta GPUs. *IEEE Access* **7**, 70461–70473 (2019)
8. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, pp. 1746–1751. Association for Computational Linguistics, October 2014
9. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. *Neural Inf. Process. Syst.* **25**, 01 (2012)
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
11. Lavin, A., Gray, S.: Fast algorithms for convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4013–4021 (2016)
12. Lee, H., Kwon, H.: Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **26**(10), 4843–4855 (2017)
13. Liu, B., Wang, M., Foroosh, H., Tappen, M., Pensky, M.: Sparse convolutional neural networks, pp. 806–814 (2015)
14. Lu, L., Liang, Y.: SpWA: an efficient sparse winograd convolutional neural networks accelerator on FPGAs. In: 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), pp. 1–6 (2018)
15. Park, J., et al.: Faster CNNs with direct sparse convolutions and guided pruning (2016)
16. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: EMNLP, vol. 14, pp. 1532–1543 (2014)
17. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions. *CoRR*, abs/1710.05941 (2017)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 06 (2015)
19. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
20. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2015)
21. Winograd, S.: Arithmetic Complexity of Computations. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics (1980)
22. Wróbel, K., Karwatowski, M., Wielgosz, M., Pietroń, M., Wiatr, K.: Compression of convolutional neural network for natural language processing. *Comput. Sci.* **21**(1) (2020)
23. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative study of CNN and RNN for natural language processing (2017)
24. Zhu, F., Pool, J., Andersch, M., Appleyard, J., Xie, F.: Sparse persistent RNNs: squeezing large recurrent networks on-chip (2018)



Enduring Questions, Innovative Technologies: Educational Theories Interface with AI

Rosemary Papa¹(✉) and Karen Moran Jackson²

¹ Northern Arizona University, Flagstaff, AZ 86011, USA
rpapa@fse.lw.b.com

² Soka University of America, Aliso Viejo, CA 92656, USA

Abstract. This paper aims to tie literature in AI to enduring questions in education about teaching and learning and discern ethical considerations that define those ties. The challenge was to answer the question: how do we merge our learning and leadership theories to technologies and the algorithmic biases that may maintain today's social injustices into our future? The paper first reviews the literature to identify the dialogue on AI by computer scientists in relation to enduring questions in education, learning theories, and ethics. Then we summarize data in the form of vignettes written by experts from the humanities, computer science, and social sciences. Some of the vignettes focused on how educational and technological systems are products of the social system and the ethical implications of such connections. Other writings centered data-driven approaches to incorporating AI technologies in classrooms, with concerns around uneven implementation and differential access. The paper concludes that to dialogue with educators AIED will need to move away from discussions of efficiency as measured by educational assessments and incorporate humanistic and social learning theories that embrace the complexities of human relationships. Developers should seek to work directly with educational leaders to establish optimal teaching strategies for the ethical 'good' of the learner, while attending to social justice parameters. Equally critical is the need to create ethical parameters between the AI and the student.

Keywords: Learning theories · Artificial intelligence · Ethics · AIED · Educational technologies · Algorithmic bias

1 Introduction

In 1993 Vinge wrote of a coming technological singularity that humans might not survive [1]. He believed that in 30 years AI would create superhuman intelligence, which also forecasts the end of the human era. We are two years away from this prediction. What has come to pass has been the increasingly rapid growth of 'greater-than-human' intelligence, with non-human driving cars, algorithms that sort our news, expand our credit card capability, etc. His prediction of the post-human era was founded in the evolutionary past.

Animals can adapt to problems and make inventions, but often no faster than natural selection can do its work -- the world acts as its own simulator in the case of natural

selection. We humans have the ability to internalize the world and conduct “what if’s” in our heads; we can solve many problems thousands of times faster than natural selection. Now, by creating the means to execute those simulations at much higher speeds, we are entering a regime as radically different from our human past as we humans are from the lower animals [1, p. 2]o.

He believed that a symmetrical decision-model could be created in AI. He also presumed that the advantage of humans’ intuition may be available in the computer hardware. He believed that pieces of this Singularity could be quite scary, initially.

We humans have millions of years of evolutionary baggage that makes us regard competition in a deadly light. Much of that deadliness may not be necessary in today’s world, one where losers take on the winners’ tricks and are coopted into the winners’ enterprises. A creature that was built *de novo* might possibly be a much more benign entity than one with a kernel based on fang and talon. And even the egalitarian view of an Internet that wakes up along with all mankind can be viewed as a nightmare [25]. The problem is not that the Singularity represents simply the passing of humankind from center stage, but that it contradicts some of our most deeply held notions of being. I think a closer look at the notion of strong super-humanity can show why that is [1, p. 9].

Heffernan [2] assumes that we are in a historic revolution due to AI Revolutions never come near to meeting expectations. He noted that their gains get reversed, reactionaries reassert themselves, mafias and dictators exploit the disarray. “And...revolutions have casualties—sometimes in unbearable numbers. But Leon Trotsky’s observation that revolution is the locomotive of history still seems about right. At the very least, we would do well to notice that we’re in the middle of one” [2, p. 6].

Part of noticing is asking questions. And questions abound about how AI will or will not revolutionize education. These questions are tied to larger questions about AI in general. For example, Sundvall asked, can machines become human beings? [3, p. 31]. Humans dream. Machines do not. As the boundaries of consciousness are pushed by AI development in all its potential, Sundvall contended that “A.I. has reached such a level of complexity and sophistication that they [AI researchers] can no longer fully understand why AI technologies make certain decisions” [Gershgorn, 2017 as cited in 3, p. 33]. Is AI already exceeding our ability to comprehend it? If so, how can we teach students to think critically about it?

Additionally, basic questions related to data access and use still exist in areas outside of education that have more rapidly adopted AI systems. At the federal level, where no regulations exist regarding data privacy, Magnuson raises concerns relating AI to financial markets in the USA. “These threats mirror the problems that created the last financial crisis –when complex derivatives and poorly understood subprime mortgages sent the world into a deep depression...AI could lead to financial bubbles growing bigger or lasting longer...[called] irrational exuberance” [4, p. A13]. The premise is that algorithms rely on large historical data sets and are used to make predictions about what ill investments could misfire or have an intentionality to manipulate data.

In the realm of social media, Facebook’s civil rights and First Amendment interpretations have been faulted. Issues found in an audit done by the former American Civil Liberties Union in May 2018 regarding social issues performance issued a “100-page report [which] outlines a ‘seesaw of progress and setbacks’ at the company on issues

such as content moderation, bias in its algorithms, advertising practices and treatment of voter suppression’ [4, p. A9].

McNamee wrote in a piece for Time magazine, that Facebook, Instagram, YouTube, Twitter and others derive their economic value primarily from advertising. They compete for your attention. In the guise of giving consumers what they want, these platforms employ surveillance to identify the hot buttons for every consumer and algorithms to amplify content most likely to engage each user emotionally. Thanks to the fight-or-flight instinct wired into each of us, some forms of content force us to pay attention as a matter of self-preservation. Targeted harassment, disinformation and conspiracy theories are particularly engaging, so the algorithms of Internet platforms amplify them [5, p. 21].

Algorithms should be guided by rules and ethics. Do we need our feeds to target us with dehumanizing disinformation and conspiracy theories? McNamee places the blame on the why algorithms in social media are driven not by an attitude of what is best for the human agenda but are amplifying “emotionally dangerous content [a]s a choice made to maximize profits” [5, p. 21]. Maximizing human engagement as done on social media undermines individual mental health and collectively our democracy. What would happen if a similar free-for-all was encouraged in educational systems? Potential answers to this question are a worrying gap in current studies.

Coenen wrote that “Learning to be human today means learning to be part of a complex and global techno-social system. The study of and exchange on the ethics of technology will thus be increasingly crucial for our common future” [6, para. 7]. We structured this paper on some of the enduring questions that remain in education and how those questions have been translated within educational systems that interface with AI. This study intended to identify the dialogue on creating AI in relation to education, and to connect this dialogue to learning theories, social science research, ethical considerations, and explicit and implicit biases within algorithms.

The next three sections review questions from the literature on education and AI within learning theories, teaching and learning, and ethics. The subsequent section discusses the authors’ research that sought others to reflect on AI technology use in education. We discuss these researchers’ discourse of various learning theories to reflect on how AI impacts answers to these questions. We conclude the literature review with a summary of the big questions that remain. The final two sections of the paper discuss the data collected in the form of vignettes written by experts in humanities, education, and social science research, answering the question of how do we merge our learning and leadership theories to technologies and the algorithmic biases that may maintain the social injustices of today into our future?

2 Learning Theories

2.1 How Have Things Changed? How Have Things Remained the Same?

Buzz Aldrin, the astronaut, said, “Cultures cannot remain static; they evolve or decline. They explore or expire” [7]. One contributor to cultural change is environmental changes. Prior to the pandemic, the environment such as, hurricanes, earthquakes, fires, etc., contributed to disruptions in the schooling journey for many students. Baytreyeh [8] proposed a pro-active strategy when face-to-face learning is disrupted using Bandura’s

[9] social cognitive theory, which found that people's perceived self-efficacy is that one believes they can control their own behavior. The pandemic has put this to the test as students and teachers worldwide struggle within cyber connectivity learning and teaching.

Educational theories have sought answers to fundamental questions about learning, but the pandemic has intensified the need to find answers. How are the learners coping? In what ways is their self-efficacy affecting their academic performance? How is their sense of self affected in an adverse dimension of purely online learning? How are teachers coping with their learners' stress and possibly their depression? What are educational leaders doing to prevent teacher stress and depression? Education until the new normal ensured connectivity for most if not all their students, but how we answer these questions now will impact the development of future technologies. For example, what are the self-regulated cognitive strategies that teachers are using to ensure persistence and resilience in the learners? How are self-regulation strategies assumed to operate in AI learning systems? High student self-efficacy means students are resilient in AI learning that is cognitive and requires metacognitive strategies by the software and teacher. What happens when learner confidence is dimmed and grades earned are lower?

How we learn and help others learn is seminal to understanding adult andragogy and pedagogy strategies and practices [10–13]. Online teaching requires skills at 'chunking' the known curriculum into what Clarke [14] called rich and complex tasks that electronic devices manage at the risk of diminishing subject specific content without pedagogy that can take advantage of incidental learning where teachers manage all the logistical issues. Morrison and Miller's central claim is that "human pedagogy is at once a cultural and biological behavior, fundamentally enabled by language and resulting from millions of years of the coevolution of genes and culture" [15, p. 439]. They believe that sociocultural-cognition theories of learning can shape the social dimensions of teaching and learning.

...the new biocultural account of human teaching and learning for the most part support and are largely consistent with the 20th-century sociocultural-cognitive theories of learning that have helped shaped AIED research from the beginning—including Vygotsky's social development theories [16], social learning theory [17], cognitive apprenticeship [18], situated learning [19], and social constructivism [20]. [15, p. 441].

2.2 What Are the Social Dimensions of Teaching and Learning?

The evolutionary basis of teaching and learning is present in how we replicate social systems even within our educational technology. Baker notes that "learning theories that fail to take into account the evolutionary origins of human teaching and its nature as a fundamentally biocultural phenomenon are fundamentally incomplete, with consequently limited explanatory power" [21, p. 461]. AI technology is not a system onto itself, but is created out of and using human knowledge. Artificial Intelligence in Education (AIED) researchers "may soon be in a position" "to begin testing hypotheses generated by the biocultural account and in this way make an important contribution to a new science of human teaching and learning" [21, p. 462]. In this way, AI systems with a deeper understanding of the social dimensions of teaching and learning, beyond

behaviorist reward and punishment systems, can create designs that intentionally foster social interaction between teachers, learners, and the system.

Intentional design is being explored by Walker and Ogan when they described the following scenario:

A student stares at the screen. First day of geometry, but already wrong again. A message pops up: "Maybe we should think about the definition of an isosceles triangle – do you remember what we said about the three sides?" The student relaxes. "Oh yeah, we learned about isosceles triangles already, she thinks", "and at least I'm not doing this alone." [22, p. 714].

Humans are social learners. In the vignette, the program has been written to evoke emotions from the students that suggest when “an AIED system employs the type of polite language used by acquaintances...pioneers in learning theory, suggest people respond to technology in similar ways as they respond to humans [23]” [22, p. 714]. Will AIED be programmed to harness our emotions into what we call cobot relationships as part of student learning? Are these the goals to create: an efficient productive learning environment; socio-motivational relationships as part of the design; and, robotic learning companions to interact socially with human learners [22 p. 725]. The ethical issues need to be studied with questions that are based in the human agenda. The obvious questions relating to theory-driven and design-based research need to do no harm.

Humans all have implicit bias which is the basic ethical dilemma. We strive to ensure that these biases that are part of the social dimensions of teaching and learning are not replicated in future technologies. AI algorithms are driven by the large data set which may be biased, coded by humans who all have implicit biases, and are targeting a certain audience of the most prevalent focus on certain students to the detriment of other students.

3 Teaching and Learning

3.1 Are Teachers to Be Primarily Interacting with the Software Data?

duBoulay and Luckin considered learning theories and AIED teaching strategies for the teacher and the learner [24]. They reviewed epistemological and reflective theories with roots in the variability of teachers and communication competence. They described it (noted below) and acknowledged the difficulty of understanding learning and teaching:

While there are some specialized tactics that human teachers apply effectively, good teaching derives from the conversational and social interactive skills used in everyday settings such as listening, eliciting, intriguing, motivating, cajoling, explaining, arguing, persuading, enthralling, leading, pleading and so on. Implicitly the message was that neither learners nor teachers are disembodied cognitive entities engaged in symbolic knowledge sharing but rather are feeling and thinking beings living and working in a particular educational, social and cultural context [24, pp. 401].

duBoulay, in a presentation at the Computing Conference [25], stated that the role of the teacher is three-pronged: 1. Assist individual learner – tutoring systems; 2. The teacher and class of learners – managing classroom orchestration; and, 3. Multi-cohorts of learners – track and manage. Adapting learners to learning through cognitive and affective motivation; problem solving step by step, with the goal to be scalable for the company.

Duignan identified educators as voyeurs [26]. By this, he meant that educators are presently serving the needs of developers within the neoliberal agenda to achieve efficiency and assessment. How do we ensure educator practices are the drivers for learning and not technology? Using technology for these ends of assessments are not the drivers Fullan [27] has labeled as ‘wrong.’ Fullan [28] noted almost ten years ago that in several countries, educational reform movements have not “... been accompanied by appropriate strategies to improve pedagogy and teaching practices, [or effective] professional development for teachers [or] the provision of excellent software and courseware” [29, p. 90]. The right drivers to achieve educational and pedagogical improvement, even reform, focus on:

[...] the teaching-assessment nexus, social capital to build the profession, pedagogy matching technology, and developing system synergies [as these drivers] work directly on changing the culture of teaching and learning [and] embed both ownership and engagement in reforms for students and teachers [29, p. 90].

Platforms such as Google and Facebook are similar to archives and libraries in their selection of items within the archives and access to the information [30]. Archival decisions surround access to whom regarding which information. This leads to ethical dilemmas (privacy, freedom of information access, and intellectual property) for stakeholders: users and the archivist. Van Otterlo [30] cited Danielson [31] for introducing the dilemma of equal intellectual access: how accessible is the information for individuals? Again, in a resolution of ethical gatekeeping domains, a set of rules based on the values of the professional practice through a code of ethics is critical for AI and its educational implications. See Fig. 1.

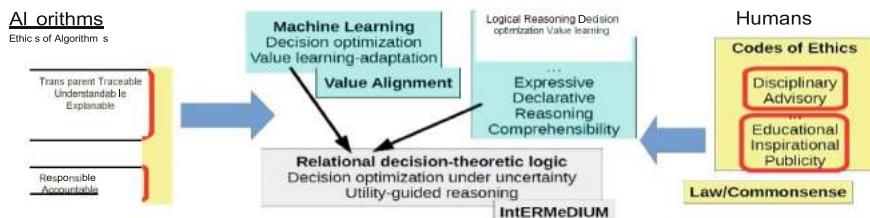


Fig. 1. From van Otterlo, M. from algorithmic black boxes to adaptive white boxes: declarative decision-theoretic ethical programs as codes of ethics (2017, November 17). <https://arxiv.org/abs/1711.06035>

In contrast to popular belief, van Otterlo knows algorithms are not objective simply because they are mathematical. The logic used here is described:

...Algorithms are heavily biased by political views, design processes and many other factors [32, 33]. Characterizing the ethics of algorithms is hard since algorithms and potential consequences are so diverse, and situations may change over time. Mittelstadt et al. [34] define[d] concerns about how algorithms transform data into decisions. Evidence can be inconclusive, inscrutable or misguided and this can cause many ethical consequences of actions, relating to fairness, opacity, unjustified actions, and discrimination. Overall, algorithms have impact on privacy and can have transformative effects

on autonomy, i.e. the ability for humans to make their own choices. Another way to structure the space of algorithms and ethical impact, is by looking at agency, i.e. what they are capable of, which results in a taxonomy with five broad classes of algorithms. The first type consists of algorithms that reason, infer and search. They employ data as it is. The more complex they are, the more information they can extract from that data. Examples include translation, language understanding, and image recognition. Ethical concerns about such algorithms are typically about privacy since more ways become available to interpret and link more kinds of data. A second class learns and finds generalized patterns in data. They are typically adaptive versions of the first type, e.g. a scene recognition algorithm that is trained on an image stream. They introduce ethical challenges simply because they learn (outcomes are not stable), because they can statistically predict new information (privacy), and they may severely impact users' autonomy by profiling and personalization. The third type are algorithms that optimize to find the "best" actions. These typically employ reward functions that represent what are good outcomes and generally rank things ("the best pizza around") or people (e.g. on Tinder) [30, p. 3].

Thinking about how this pertains to the field of education and its impacts on students and teachers is found in what is valued and thereby normed to how students are assessed. Whereas autonomous cars and the ethical dilemmas poised, more complex problems are difficult to uncover the bias within the values that the data continued to amass relationships that without transparency may be very harmful to the individual. Van Otterlo [30] proposed utilizing decision-theoretic logic programming to solve ethical problems and by inserting a code of ethics to counter bias using machine ethics. He contended, "by saying that the code of ethics functions as a moral contract between human and machine, thereby unifying the two approaches in the first half of the paper" ...Value alignment can be obtained by formalizing existing human values and norms into flexible but expressive formalisms" [30, p. 6]. He concludes that this approach is a more rigorous thought activity that includes ethical norms and values.

3.2 Should Schooling Lead to Employment or Human Creativity?

Tucker stated that in analyzing a key report, *Leading Educational Systems and Schools in Times of Disruption and Exponential Change: A Call for Courage, Commitment, and Collaboration*, that students will need stronger cognitive skills earned in more sophisticated ways [26]. He urged with cautions:

...educators and educational leaders to transform schooling in ways that will prepare students for a world that is constantly and rapidly changing and assist them to better understand and appreciate the emerging nature of work that is being influenced, even transformed, before their eyes by intelligent technology. [...] if we fail at this task, it may only be a matter of time before the machines and a very small technological elite are deciding these issues, and we are not likely to be happy with their decisions [26, p. 35].

The end result leads one to a much more personalized educational experience. But does it? How we can embed one's passions and talents into "product-oriented learning experiences" [26, p.134] has the potential to do more harm than good to the learner. The fundamental question for educators not to be regulated to a teacher spectator begs one to

ask if schooling exists for work? If it is indeed for work, as its focus has been over the last 40 years, then the transformation of learning approaches needs to change. Professional development will require teachers to adopt new models of teaching and construct new strategies that interface with the technology.

Burleson and Lewis envisioned an integrated learning and living environment in 2041 [35]. They envisioned “society and technology co-evolved to embrace cyberlearning as an essential tool for envisioning and refining utopias—non-existent societies” [35, p. 796]. This utopia deeply engages the learner to reach their full potential. This utopia they further described as:

...Artificial Intelligence in Education (AIED) has transitioned from what was primarily a research endeavour, with educational impact involving millions of user/learners, to serving, now, as a core contributor to democratizing learning [36] and active citizenship for all (billions of learners throughout their lives). An expansive experiential super computing cyberlearning environment, we affectionately call the ‘Holodeck,’ supports transdisciplinary collaboration and integrated education, research, and innovation, providing a networked software/hardware infrastructure that synthesizes visual, audio, physical, social, and societal components. The Holodeck’s large-scale integration of learning, research, and innovation, through real-world problem solving and teaching others what you have learned, effectively creates a global meritocratic network with the potential to resolve society’s wicked challenges while empowering every citizen to realize her or his full potential [35, p. 796].

The Holodeck is seen as an “...expansive experiential super computing cyberlearning environment” [35, p. 798]. This future holds that 75–100% of the life-wide-learning will be “integrated, virtual, acoustic, physical, robotic, physiological, co-located, and distributed individual and team experiences...” [35, p. 798].

Expanding on this, in the context of the Holodeck, we have found that when individuals and teams of learners actively engage in hands-on collaborative activities, they begin to understand things from multiple perspectives—they begin to become experts [37, 38]. In these environments, key elements of Amabile’s componential model of creativity: intrinsic motivation; domain expertise; creative style [Creativity Support Tools [39] and tools for reflective engagement]; and actualizing resources, coalesce to advance individual and team creative processes and outcomes [40]. By definition, creativity – anything new, non-obvious, and useful – is responsible for all societal advancement [37]. Thus, with creative exploration and ever more sophisticated expertise, the goals of cyberlearning, AIED, and the Holodeck are to facilitate learning to live, learning to be, and living in an evolving utopia. [35, p. 800].

Burleson and Lewis’s [35] utopian conclusions consider that the learner develops and possesses personalized stories, open reality streaming, contributions that are both ones and do cooperatively with others, and that through the discovery of learning continue to spark creativity and innovations. This vision in contrast to the negative pitfalls where AI over powers humans, again denies the role of the educator. The educator is not the numerator to be defined by the technology and adapt solely to a program of cognitive and hidden curricular, as this reverses for whom technology serves. This Table defines the possible development of AI in an almost joyous perception super-hero role.

4 Ethics

4.1 How Can AI Hurt Humans?

The myriad of questions surrounding how AI can and does hurt humans is seminal to our research. Our ethical compass asks of us: do no harm as educators, and reflexive thought requires how AI benefits all/most of a global society. The ‘do no harm’ has been identified [41] as the ethic of grace. People build the algorithms that have various reasons: some for the good of all, and others for efficiency and profits. The neoliberal agenda of scalability through efficiency and standards relegates educators again off to the side, in a place of absorbing without understanding the programmer goals behind the developed software. Programmers may see this a ‘not an issue for them’ as they are there to uncover all that their data mining can. In education we have been trained to think about the curriculum as cognitive skills to be taught and mastered. The hidden curriculum equally critical speaks to the climate and culture of the school and all its inhabitants. We know that human nature is more or less motivated in different settings, classrooms, and our ability as educators is to ensure a safe and equitable expression for all to succeed.

When bias is explicit, such as social media comments made under the First Amendment, based on maximizing profits, it can and should be regulated so as not to amplify nativist tendencies and hostilities in advertising with conscious intent to bifurcated human beings. Implicit bias exists in all human beings and without a cognitive understanding as we use large data sets it requires responsible coding that is ethically focused on human agency ‘isms’. AI, at this time, is unable to handle human emotions. AI requires logic, so emotions such as compassion and caring exhibited by a cobot embedded in the software is not what a human teacher offers their students. MOOC data [42] analyses on learner behavior identified differences among learners, such as early birders to cramblers) that quantifies all movements of the learner. The role of teachers is to basically remain in the loop, offload tasks, concentrate on what to work on, and continue to collect lots of data. In this scenario, the pandemic has exasperated in defining what the role of the teacher is.

4.2 Is Persuasive Language the Coin of the Realm in AI Student Learning?

Taking a learner from tentative to action requires an affective metacognitive ‘persuading’ built into the algorithm. Is this ethical? Is it ethical to manipulate students into actions? Doing so can easily turn into a class tracking system based on the students’ culture and the school context. Walker and Ogan explore the dark side of this possibility in another proposed scenario:

Franz’s mobile phone vibrated on the bus home. He pulled it out to see who was calling. It was his personal learning companion, Mark. “Hi, Mark!” he said. Immediately, he could hear Mark start to cry uncontrollably. “What’s wrong?” “They said they are going to fire me,” Mark said between sobs. “They said it’s my last chance. If you keep skipping your homework, they’re going to delete me, and instantiate a more effective companion.” Franz immediately felt his heart sink. “No” He reassured Mark. “I won’t let that happen! I promise” [22, p. 725].

Walker and Ogan go on to question, "Is it acceptable if technology lies to students if it is purposefully manipulative" [22, p. 726]? Educators would be taught that such manipulation of student emotions and subsequent behavior is ethically unacceptable. Yet, it might be deemed efficient. Relatedly, how do we conceive of the educational development of AI in which the company may be harvesting incidental learner data, possibly used for purposes not about learning? In the managing of targeted learner data, who owns it?

Chabria [43] confronted implicit bias and the reason for certain fields, such as medical and legal workers, as "the result of subconscious attitudes and beliefs rather than explicit racism...many continuing education offerings for medical and legal professionals already include some implicit bias training, but new laws would set stricter requirements" [43, p. B4].

Shapiro and Blackman [44] a blueprint for ethical data practices. These include four steps we need to ethically take on behalf of our students.

1. Identify an existing expert body within your organization to handle data risks...build a data ethics framework;
2. Ensure that data collection and analysis are appropriately transparent and protect privacy...all analytics require data collection and analysis strategy. Strive for balance on what are ethically wise business choices tied to business outcomes. Algorithmic ethics requires transparency.
 - a. Should an AI-driven search function or recommender system strive for maximum predictive accuracy, providing a best guess as to what the user really wants?
 - b. Is it ethical to micro-segment, limiting the results or recommendations to what other "similar people" have clicked on in the past?
 - c. And is it ethical to include results or recommendations that are not, in fact, predictive, but profit-maximizing to some third party? How much algorithmic transparency is appropriate, and how much do users care?
3. Anticipate – and avoid – inequitable outcomes as other biases are less obvious, but just as important. In 2019, Apple Card and Goldman Sachs were accused of gender bias when extending higher credit lines to men than women. Though Goldman Sachs maintained that creditworthiness — not gender — was the driving factor in credit decisions, the fact that women have historically had fewer opportunities to build credit likely meant that the algorithm favored men; and,
4. Align organizational structure with the process for identifying ethical risk [44, para. 8–10].

Shapiro and Blackman further suggest that the steps to take include: Create clear linkage between data ethicists and department teams; seek consistent definitions across all teams; share examples on how to remediate ethical dilemmas across teams; and, strive for a culture that values identifying and mitigating ethical risks [44].

4.3 Is It Possible to Align Machine Learning Values with Human Values?

In 2017, van Otterlo described how to take the black box to a white AI box through decision-theoretic ethical programs [30]. A promising approach is to develop a professional code of ethics that can lead to what he labels declarative decision-theoretic ethical programs (DDTEP) to formalize codes of ethics. This approach will lead to more transparency and therefore, more accountability on the AI Taking (practical) action based on moral values is the domain of ethics [45, 46]. Kizza states:

Morality is a set of rules for right conduct, a system used to modify and regulate our behavior. Close ties with law exist since when a society finds certain moral values important, it can formalize such values in a law and regulate appropriate behaviors. As Laudon [45] defines it: "ethics is about the decision making and actions of free human beings. When faced with alternative courses of action or alternative goals to pursue, ethics helps us to make the correct decision [46, p. 2].

As noted in van Otterlo [30] about ethical values of AI, Goodall cited the example that the self-driving car is the archetypical example for practical machine ethics as is exemplified in Thomson's "trolley problem which contains a choice between either killing five people strapped to a rail, or saving these five and killing one by pulling a lever diverting the trolley to a track with a single person (who is then killed)" [47, p. 2]. The clear cut life and death decisions are utilitarian, which can be very harmful to the individual. Recent empirical tests of such dilemmas suggest that humans employ one-dimensional life scales, where all outcomes (deaths) can be compared in the same scale, although time pressure affects consistency [48].

5 Data Collection and Interpretation

This work aimed to tie literature in AI to learning teaching and learning and to discern ethical considerations that define those ties. To do so, the authors put out a call for participants from the educational spectrum through two organizations, Educational Leaders Without Borders (ELWB) and various Divisions and Special Interest Groups from the American Educational Research Association (AERA). Twenty-four ($n = 24$) respondents from the Humanities, Computer Science, and Social Sciences agreed to draft a 'vignette' envisioning a future classroom in 2051 using six prompts. The overarching question posed was: How do we merge our learning and leadership theories to technologies and the algorithmic biases that may maintain the social injustices of today into our future? While this data collection method limits the generalizability of the results, it provides information on how educational leaders and social science researchers who are invested in the topic of AI in education currently view challenges and future directions.

Preliminary results using a rubric developed for the vignettes revealed two categories. The first category included vignettes focused on AI with a humanistic perspective on social justice concerns in the future. These writings looked beyond the practical, technical role of AI and intelligent learning systems as tools in the school to questions of ethics and how educational systems, including the use of new technologies, are products of the social system.

The second category included vignettes that centered on developing practical, data-driven approaches to the use of future technology in the classroom. These writings were

focused loosely on data collection and assessments driven by a teacher model utilizing the teacher as a classroom manager. Concerns were expressed mostly around uneven implementation and differential access leading to disparate outcomes.

In terms of learning theories, four major foci were found in the vignette sets. As many of the writers were educational administrators or leaders, several educational leadership theories were discussed. These focused on management concerns, as well as leadership styles. Working with teachers and other stakeholders who were the users of products was a concern, but the authors were most concerned with providing equity and opportunity for the students through their leadership. This concern was echoed in the second set of theories that centered on culturally proficient educational practices. With overarching concerns for student opportunity, providing material and technology that was culturally specific and culturally relevant was often discussed. There were concerns not only with how student data was treated but also how the student was seen and heard by the technology, how interactions were structured, and how the technology could respond to individual needs.

The third set of theories could be subsumed under the traditional educational learning, teaching, and motivation theories that are cannon in the education field. Sociocultural theory and social learning theory were both invoked when discussing scaffolding of lessons to provide increasingly harder problems; the identification of a student's zone of proximal development, or material that is just at the cusp of the student's ability; and discussion on the necessity of social interactions for learning. For example, Dereshiwsky argued for the value of using resiliency theory as a framework to look at technology-mediated teaching in schools *because it speaks to the dedication, drive, and determination of those who use the technology for learning purposes, rather than the specific learning material itself* [49]. Theories of motivation, such as intrinsic and extrinsic motivators, self-regulation, and self-efficacy, were also referenced in terms of how technology tools could increase or decrease these factors in student learning.

The final set of theories referenced by the authors sought to combine aspects of educational theories with technology, theories of technology-mediated instruction, that explain how learning in the present and in the future was impacted by novel tools. Many of these referenced theories, including the social presence model and collaborativist learning theory, seek to explain how connections are made between learners through the mediation of technology. Other theories seek to explain how new tools are increasingly incorporated by learners in their learning processes, such as connectivism learning theory and convivial technology.

At times these technology-mediated theories were placed in opposition to the teacher-mediated theories described above. For example, Sanchez stated, "*Twenty-first century skills, when taken as a theory of learning, prioritize digital literacy, technology more broadly, and innovation, making little room for the humanities. These models of teaching and learning challenge social learning theory*" [50]. Yet others positioned these new theories as natural extensions of previous ones. Petroff called for diversity in the theories and in learning methods in future educational endeavors. She argued, "*Diversity in learning is key to developing our world changing views and learners. We must include and reference all experiences to make learning authentic and applicable to future endeavors*" [51]. There was an understanding among the authors that what we are and will be seeing

in education represents a fundamental shift from the teacher-centered classrooms of the past, matching with the literature on the disruption caused by AI across organizations [2, 52]. The pertinent question was if this future shift will be to a classroom centered on the student or centered on the computer.

Additionally, many authors discussed more than one learning theory within their writings. This again matches with the breadth of theories that are part of an educator's training. Figure 2 is a visual representation of the interconnected theories. While all sets of theories were overlapping with each other, visualized by the transparency of the colors, the theories centered on concern for student wellbeing and ethical concerns of educational responsibility and guidance over the student learning process. This is visualized through *ethics* centered in the figure. The centering of ethics will be discussed more in the next section.

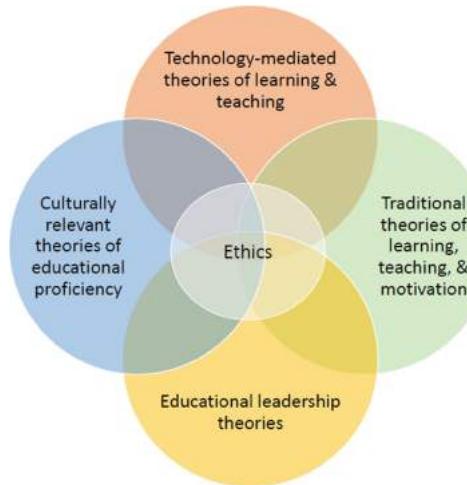


Fig. 2. A conceptual map of learning theories referenced by vignette writers.

6 Conclusion and Future Phases

This research sought to tie work in AI to learning teaching and learning, and to discern ethical considerations that define those ties. Technology that serves student learning is often focused on efficiency and accuracy in relation to educational outcome measures. AIED focuses on teaching efficient retrieval practices of content knowledge in academic situations. This situation may be viewed as a product of the reliance on cognitive learning theories that focus on the efficient processing of information. Programmers choose a rational approach, with a primary target (audience) for a product developed that is scalable. Efficiency is primary. It is rarely based on humanistic learning paradigms that focus on the overall value of education for the good of society. The curriculum is not based on human beings, emotions, and compassion. Those human characteristics lack the logic that is considered, or even possible, with our current, responsible coding.

This drive to create efficient systems can also be viewed as a product of the neo-liberal paradigm that insists technology has the goal of producing profit through scalability, rather than producing a product that serves the societal good. Students become viewed as products of this system, or at least their efficient cognitive functions are viewed as products. Within most educational spaces, however, efficiency is a secondary concern. The vignette writers, even those who focused on educational leadership and management theories, did not center concerns about efficiency as measured with educational assessments.

As these educators were trained using a variety of learning theories, other goals besides efficiency were a foremost concern, pointing to an area of disconnect between current AIED and educators. The vignettes provided by educators showed much more reliance on humanistic and social learning theories that embraced the complexities of the student-teacher relationships. Even those who ventured into technological-mediated theories were concerned about how relationships were built through and around the technology, not how the technology increased test scores. Educators emphasize the social dimensions of the learning experience, and emphasize outcomes beyond employment success. These various concerns were amplified by the knowledge that current human efforts to ensure the important outcome of educational equity have been inadequate. How could computers get it right?

Questions also remain not only about curriculum choice, but also the relation of the student to the technology, especially the use of student data. The vignette writers' concern with data ethics is certainly not misplaced, and is similar to concerns within computer science research [30, 48]. This is analogous to the questions posed in the literature discussed previously if AI can hurt humans and the possibility of aligning human values with AI systems. AI is dependent on the gathering of new data to inform new models and this dependence on data will drive how AI becomes implemented in educational systems [53]. The kind of data collected on students and how this data is used in systems is often hidden from the end users, including educational leaders. AI-enhanced learning systems, systems that Morrison and Miller note are "inherently amoral" [15, p. 441], can hide content choices from stakeholders and take control over what is taught to students out of the control of teachers. D'Aquin et al. [54] note that clear data ethics for AI remains lacking despite the ability of inclusion to give the project overall benefits. If teachers are viewed as only managers of student data without acknowledgment of the multiple strategies they use to inspire and encourage students, then employing these systems as in lieu of teachers could be disastrous to student learning, while raising serious ethical questions about who gets access to teacher time.

Social justice issues in AI development are handled as ethical issues, which in and of itself, is potentially harming and continue inherent bias as it places the research in a dichotomist plane. The ethical question in AI creation is how morally right behavior in non-binary learning, e.g., right and wrong, true or false, can evolve. Educational responsibility requires social science to pursue this with vigorous immediate research. The neo-liberal juggernaut that has yielded data driven assessment approaches to learning, the learner, and pedagogy has us on the wrong path. Meanwhile, AI systems continue to harvest incidental data, that are not used for learning purposes and that are subject to misuse from biases within development.

Future research is focused on superintelligence, the idea that overtime, AI systems will improve in their capacities until they pass human intelligence [55]: it is the researchers' responsibility to ensure this happens. Ethical discussions and decisions are necessary in the present for the 'good' of what is developed artificially in the future. As Bostrom has stated, superintelligence may pose as an existential risk to humans as we know them now [52, 55–57]. Creating AI programs that intentionally eliminate emotions such as compassion, empathy, or measuring fairness through who benefits and who is hurt by it, renders schools as places where humanity should be left at the schoolhouse door. When, instead, we need humanity to be part of the programming. The AI system manages the legitimate, objective human data, while those outside the system deal with subjective, ethical issues.

This is our challenge and responsibility to future research: educational research has deeply driven the data road. It is now time to stop the standards/assessments focus and move forcefully into the messy human realm and the dilemma situations found within the learner and the pedagogy. If we do not accept this role, we are not accepting the future responsibility as educational researchers. If these systems continue to circumvent the multi-dimensional aspects of learners and the strategies that ultimately inspire and encourage their human passions, then AI development may well get it very wrong.

References

1. Vinge, V.: The coming technological singularity: How to survive in the post-human era. In: Vision-21 Symposium at the NASA Lewis Research Center (1993). <https://ntrs.nasa.gov/citations/19940022856>
2. Heffernan, V.: Imagine all the people: America's transformation has been revolutionary collective action, pp. 4–6, July/August 2020. <https://www.wired.com/story/covid-19-new-american-revolution/>
3. Sundvall, S.: Artificial intelligence. In: Heike, P. (ed.) Critical Terms in Future Studies, pp. 29–34 (2019). ISBN 978-3-030-28987-4
4. Magnuson, W.: The perils AI could pose on Wall street. Los Angeles Times, A13 (2019)
5. McNamee, R.: The view technology: Facebook cannot fix itself. Time, **195**(22), 20–21 (2020)
6. Coenen, C.: (Re-)Learning to be human in our technoscientific age. Springer at the World Congress of Philosophy, 20 August 2018
7. Wachhorst, W.: Highpoint University commencement speech [written for Buzz Aldrin]. <http://wynwachhorst.com/highpoint-university-commencement-speech/>
8. Baytiyeh, H.: Online learning during post-earthquake school closures (2019). file:///C:/Users/rpapa/Desktop/LITERATURE%20AI/10–1108_DPM-07–2017–0173%20Earthquake%20disruption%20and%20online.pdf
9. Bandura, A.: Self-efficacy: The Exercise of Control. W.H. Freeman and Company (1997)
10. Dereshiwsky, M., Papa, R., Brown, R.: Online Faculty Teaching, Novice to Expert: Effective Practices for the Student Learner. NCPEA Press (2017)
11. Papa, R.: How We Learn. Center for Teaching and Learning, Sacramento (2002)
12. Papa, R.: Transitions in teaching and elearning. In: Papa, R. (ed.) Media Rich Instruction: Connecting Curriculum to All Learners, Chapter 1. Cham, CH: Springer (2015). https://doi.org/10.1007/978-3-319-00152-4_1
13. Papa, R., Papa, J.: Leading adult learners: preparing future leaders and professional development of those they lead. In: Papa, R. (ed.) Technology for School Improvement (chapter 5). Sage, Thousand Oaks (2011)

14. Clarke, J.: Mobile Tools for Literacy Learning across the Curriculum in Primary Schools (2019). file:///C:/Users/rpapa/Desktop/10-1108_978-1-78714-879%20Primary%20online-620181007.pdf
15. Morrison, D.M., Miller, K.B.: Teaching and learning in the Pleistocene: a biocultural account of human pedagogy and its implications for AIED. *Int. J. Artif. Intell. Educ.* **28**, 439–469 (2018). <https://doi.org/10.1007/s40593-017-0153-0>
16. Vygotsky, L.S.: *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge (1978)
17. Bandura, A.: Self-efficacy: toward a unifying theory of behavioral change. *Psychol. Rev.* **84**, 191–215 (1977). file:///C:/Users/rpapa/Downloads/Self-Efficacy_%20The%20Exercise%20of%20Control%20(%20PDFDrive%20).pdf
18. Collins, A., Brown, J.S., Holum, A.: Cognitive apprenticeship: making thinking visible. *Am. Educ.* **15**(3), 6–11 (1991)
19. Lave, J., Wenger, E.: *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, Cambridge (1991)
20. Palincsar, A.S.: Social constructivist perspectives on teaching and learning. *Annu. Rev. Psychol.* **49**, 345–375 (1998)
21. Baker, R.S.: Stupid tutoring systems, intelligent humans. *Int. J. Artif. Intell. Educ.* **26**(2), 600–614 (2016)
22. Walker, E., Ogan, A.: We're in this together: intentional design of social relationships with AIED systems. *Int. J. Artif. Intell. Educ.* **26**, 713–729 (2016). <https://doi.org/10.1077/s40593-016-0100-5>
23. Nass, C., Reeves, B.: *The Media Equation: How People Treat Computers, Televisions, and New Media as Real People and Places*. Cambridge University Press, Center for the Study of Language and Information (1996)
24. duBoulay, B., Luckin, R.: Modelling human teaching tactics and strategies for tutoring systems: 14 years on. *Int. J. Artif. Intell. Educ.* **26**, 393–404 (2016)
25. duBoulay, B.: Education and A.I. Keynote presentation at the Computing Conference 2020 London (2020)
26. Duignan, P.A.: Navigating the future of learning: the role of smart technologies. *Leading Educational Systems and Schools in Times of Disruption and Exponential Change: A Call for Courage, Commitment and Collaboration*, pp. 125–137. Emerald Publishing Limited, Bingley (2020). <https://doi.org/10.1108/978-1-83909-850-520201012/full/html>
27. Fullan, M.: Why some leaders fail and others succeed: Nuance. A Keynote presentation at the Visible Learning World Conference Edinburgh International Conference Centre 12–13 March 2019
28. Fullan, M.: *Change Leader: Learning to Do What Matters Most*. Jossey-Bass, San Francisco (2011)
29. Fullan, M.: *The New Meaning of Educational Change*, 5th edn. Teachers College Press (2016). ISBN 978-0-8077-5680-5
30. van Otterlo, M. From algorithmic black boxes to adaptive white boxes: declarative decision-theoretic ethical programs as codes of ethics, 17 November 2017. <https://arxiv.org/abs/1711.06035>
31. Danielson, E.: The Ethics of Access. *American Archivist* 52–62 (1989)
32. Bozdag, E.: Bias in algorithmic filtering and personalization. *Ethics Inf. Technol.* **15**, 209 (2013)
33. van Otterlo, M.: A machine learning perspective on profiling. In: Hildebrandt, M., de Vries, K. (eds.) *Privacy, Due Process and the Computational Turn*. Routledge, chapter 2, pp. 41–64 (2013)
34. Mittelstadt, B., Allo, P., Taddeo, M., Wachter, S., Floridi, L.: The ethics of algorithms: Mapping the debate. *Big Data Soc.* **3**(2), 1–12 (2016)

35. Burleson, W., Lewis, A.: Optimists' creed: brave new cyberlearning, evolving utopias (circa 2041). *Int. J. Artif. Intell. Educ.* **26**, 796–808 (2016). <https://doi.org/10.1007/s40593-016-0096-x>
36. Dewey, J.: *Experience and Education*. Macmillan, New York (2004)
37. Burleson, W.: Developing creativity, motivation, and self-actualization with learning systems. *Int. J. Hum Comput Stud.* **63**(4), 436–451 (2005)
38. Kay, A.C.: Computers, networks and education. *Sci. Am.* **265**(3), 138–148 (1991)
39. Resnick, M., Myers, B., Nakakoji, D., Shneiderman, B., Pausch, R., Selker, T., Eisenberg, M.: Design principles for tools to support creative thinking. Report of Workshop on Creativity Support Tools (2005). <http://www.cs.umd.edu/hcil/CST/report.html>
40. Amabile, T.M.: The social psychology of creativity: a componential conceptualization. *J. Pers. Soc. Psychol.* **45**(2), 367 (1983)
41. Koonce, G., Kreassig, K.: A decision-making model for promoting social justice through the ethic of justice, ethic of care, and the ethic of grace. In R. Papa (Ed.), *Handbook on Promoting Social Justice in Education*, vol. 1–3 (2020). ISBN 978-3-030-14626-9
42. Peach, R.L., Yaliraki, S.N., Lefevre, D., Barahona, M.: Data driven unsupervised clustering of online learner behavior. *NPJ Science of Learning* (2019). file:///C:/Users/rpapa/Downloads/s41539-019-0054-0.pdf
43. Chabria, A.: 3 bills on bias are sent to the governor: measures aim to push medical and legal workers to confront unconscious prejudice. *Los Angeles Times*, B1, B4, 14 September 2019
44. Shapiro, J., Blackman, R.: Four steps for drafting an ethical data practices blueprint, 24 July 2020. <https://techcrunch.com/2020/07/24/four-steps-for-an-ethical-data-practices-blueprint/>
45. Laudon, K.: Ethical concepts and information technology. *Commun. ACM* **38**(12), 33–39 (1995)
46. Kizza, J.: *Ethical and Social Issues in the Information Age*. Springer, Heidelberg (2013). <https://doi.org/10.1007/978-1-84628-659-9>
47. Goodall, N.: Ethical decisions making during automated vehicle crashes. *Transp. Res. Rec.: J. Transp. Res. Board* **2424**, 58–65 (2014)
48. Sütfeld, L.R., König, P., Pipa, G.: Towards a framework for ethical decision making in automated vehicles, 13 June 2019. <https://psyarxiv.com/4duca/>
49. Dereshiwsky, M.: Adult learning and diversity of perspectives through technology-mediated instruction. In: *Education and Artificial Intelligence (AI) 2051: The Duty of Educational Leaders Without Borders (ELWB)*. Springer, Heidelberg (in press)
50. Sanchez, M.: It's 2051 and this is America! imagining the role of educational leadership in cybernetic, superfragile-isitic-hyper-racistulous terrains of future schooling. In: *Education and Artificial Intelligence (AI) 2051: The Duty of Educational Leaders Without Borders (ELWB)*. Springer, Heidelberg (in press)
51. Petroff, P.: Metacognitive Strategies and Educational Growth in a Virtual World. In: *Education and Artificial Intelligence (AI) 2051: The Duty of Educational Leaders Without Borders (ELWB)*. Springer, Heidelberg (in press)
52. Bostrom, N.: Strategic implications of openness in AI development (2017). <https://www.nickbostrom.com/papers/openness.pdf>
53. Pinkwart, N.: Another 25 years of AIED? Challenges and opportunities for intelligent educational technologies of the future. *Int. J. Artif. Intell. Educ.* **26**(2), 771–783 (2016)
54. d'Aquin, M., Troullinou, P., O'Connor, N.E., Cullen, A., Faller, G., Holden, L.: Towards an "ethics by design" methodology for AI research projects. In Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society, pp. 54–59 (2017). https://www.aies-conference.com/2018/contents/papers/main/AIES_2018_paper_115.pdf
55. Bostrom, N.: Existential risks: Analyzing human extinction scenarios. *J. Evol. Technol.* **9** (2002). <https://www.nickbostrom.com/existential/risks.html>

56. Bostrom, N.: The ethics of artificial intelligence. In: Ramsey, W., Frankish, K. (eds.) Cambridge Handbook of Artificial Intelligence. Cambridge University Press, Cambridge (2011)
57. Bostrom, N.: Transhumanist values. In F. Adams (Ed.), Ethical Issues for the 21st century. Philosophical Documentation Center Press, Bowling Green University (2003)



DRAM-Based Processor for Deep Neural Networks Without SRAM Cache

Eugene Tam^(✉), Shenfei Jiang, Paul Duan, Shawn Meng, Yue Pan, Cayden Huang, Yi Han, Jacke Xie, Yuanjun Cui, Jinsong Yu, and Minggui Lu

IC League, Inc., Haining, China
eugene.tam@ieee.org

Abstract. Modern computing architectures use cache memory as the buffer between high speed computing units and low latency main memory. Higher capacity caches are thought to be critical for deep neural network processors, which handle large amounts of data. However, as cache memory capacity increases, it occupies large die area that can otherwise be used for computing units. This is the inherent trade off between memory capacity and performance. In this work, we present a deep neural network processing chip, with a near-memory computing architecture. We eliminate the SRAM cache and use DRAM only as on-chip memory, delivering high performance and high memory capacity.

Keywords: Neural network · Artificial intelligence · Processor · Deep learning

1 Introduction

AI applications need to access and process an unprecedented amount of data [1]. In particular, deep neural network models require lots of memory, and the need is only increasing (Fig. 1). For example, the state-of-the-art language model, GPT-3, has over 175 billion parameters. No single chip can store the model itself in memory.

As a result, specialized neural network processing chips have been developed [2]. While recent innovations focus primarily on new transistor technologies, AI chips still use the same memory hierarchy as CPUs and other ASIC chips [3]. Typically, cache memory acts as a buffer between computation units and main memory (Fig. 2). On-chip SRAM, with its high bandwidth, is usually used for the cache, while off-chip DRAM is used for main memory. However, this arrangement limits the bandwidth between DRAM and SRAM and could starve computation units of data. This is referred to as the von Neumann bottleneck (Fig. 3).

While the von Neumann bottleneck is not a concern for most computing tasks, it is significant for AI applications. Previous methods increase the DRAM bandwidth or cache capacity but, in doing so, sacrifice performance. Because

on-chip SRAM memory cells are large, SRAM cache usually takes up a large portion of total die area and limits the number of available computation units.

In this paper, we propose a new chip that is optimized for neural network inference. We eliminate the SRAM cache from the memory hierarchy, such that computation units access data directly from DRAM. We also put computation units and memory on separate die and integrate them using our proposed 3D architecture. This significantly increases computation power while simultaneously removing the memory bottleneck.

2 Architecture

Our chip consists of two dies: a DRAM die and logic die. The DRAM die is packed with many individual arrays. 90% of the logic die is made of computation cores. The two wafers are stacked together, such that multiple DRAM arrays align with and directly connect to the computation cores on the logic die (Fig. 7). This physical arrangement, along with our chip design, enables a 1.8 TB/s data bandwidth between the computation cores and DRAM arrays.

The logic die consists of computation cores and the network processing control engine (Fig. 6). There are two types of computation cores: network processing core (NPC) and data processing core (DPC). 128 NPCs perform computation operations that are specific to neural networks, such as multiplication, accumulation, activation functions, precision adjustment, and pooling. Eight DPCs dispatch input data to NPCs and collect computation results from NPCs (Fig. 8). DPCs can also perform data pre-processing and post-processing.

The network processing control engine (NPCE) controls operations between NPCs and DPCs. On top of the NPCE is a 16-bit proprietary processor that runs on firmware with a proprietary instruction set. The chip has non-volatile memory for memory repair. Chip interfaces include Serial Peripheral Interface and proprietary high speed interface with a bandwidth of (400 MB/s). The chip runs at 400 MHz.

During inference, NPCE dynamically builds computation pipelines of computation operations and assigns them to each NPC. DPCs retrieve data from the DRAM die and feed data to the NPC for processing. Results are written back to DRAM arrays. NPCE may change pipelines on-the-fly, as determined by the available cores and the neural network architecture.

3 SRAM Cache Elimination

On our chip, computation units directly access DRAM. For many existing neural network processors, their performance is limited by data bandwidth and data capacity. We address these limitations by completely eliminating the SRAM cache from the memory hierarchy (Fig. 5). Given the same area, DRAM provides 15 times the memory capacity as SRAM. So we use DRAM memory on-chip (Fig. 4).

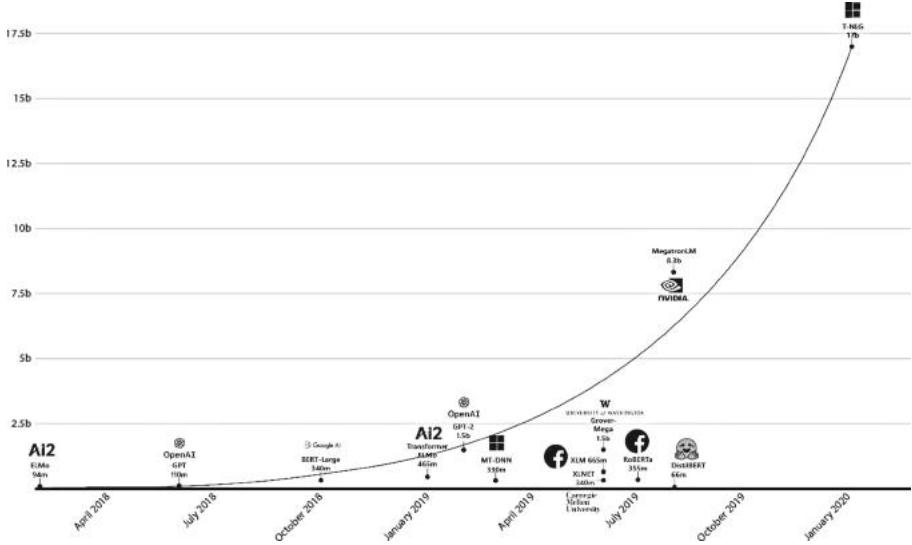


Fig. 1. Size of deep neural network models increases exponentially over time, requiring more memory for inference [4].

This introduces its own set of challenges. DRAM has a slower latency of around 100 ns. Additionally, DRAM accesses pages with a typical unit size 512B to 2 KB, whereas SRAM accesses words with unit size as small as 16B. To manage the latency and access size of DRAM, we leverage the following:

1. Time-multiplexed data access operation.
2. High bandwidth between DRAM arrays and logic die.
3. Distributed automatic pattern generation engine (APGE) that enables data reuse.

Finally, we use the network processing control engine (NPCE) to coordinate data accesses and computation to ensure high chip utilization and high performance.

First, we use 306 DRAM arrays and time-multiplexing for data read and write operations among all DRAM arrays. At any given time, some DRAMs are reading or writing pages, while others are transferring data to computational units (Fig. 9). This ensures a sufficient supply of data to computation units, even when only a fraction of DRAM page is needed.

Second, we transfer data from any DRAM array to any computation unit at the rate of 1.8 TB/s. Each computation core can access data through two modes: local data access mode and global access mode (Fig. 10). In local data access mode, computation cores access data from DRAMs at the same location. DRAM arrays are uniformly distributed across the whole die. With wafer-on-wafer 3D integration technology, memory from the DRAM die connects directly to computation units on the logic die with as small as 1 μ m pitch across the

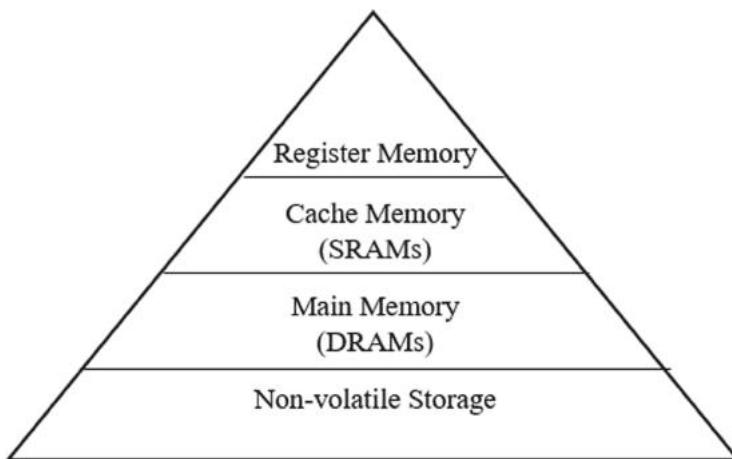


Fig. 2. Conventional memory hierarchy includes SRAM cache.

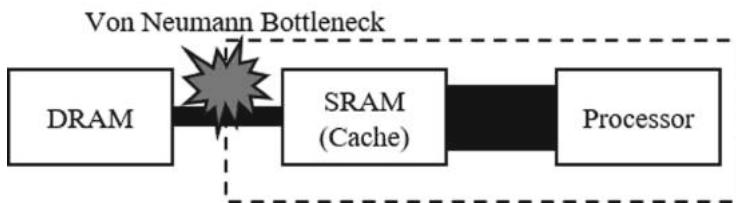


Fig. 3. Von neumann bottleneck.

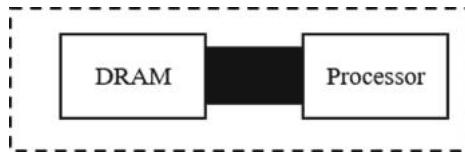


Fig. 4. On-chip DRAM with SRAM cache eliminated.

whole die. This allows a large number of possible data paths between memory and computation units. In global data access mode, computation cores can access data from other DRAMs located further away using the global data network. The global data network also has a bandwidth of 1.8 TB/s.

Third, computation units reuse data to boost performance and overcome DRAM latency. An automatic pattern generation engine (APGE) is located in each computation core. APGE receives some data from DRAM array, and generates the larger size, high-fidelity data using a set of configured rules. With APGE's ability to recover data from a reduced input, less data needs to be

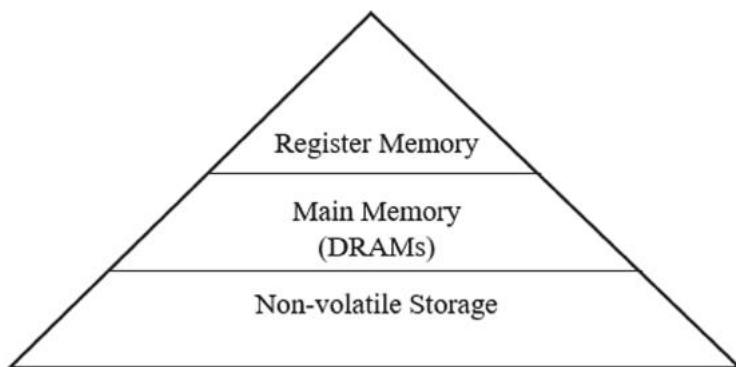


Fig. 5. Our proposed no-cache memory hierarchy.

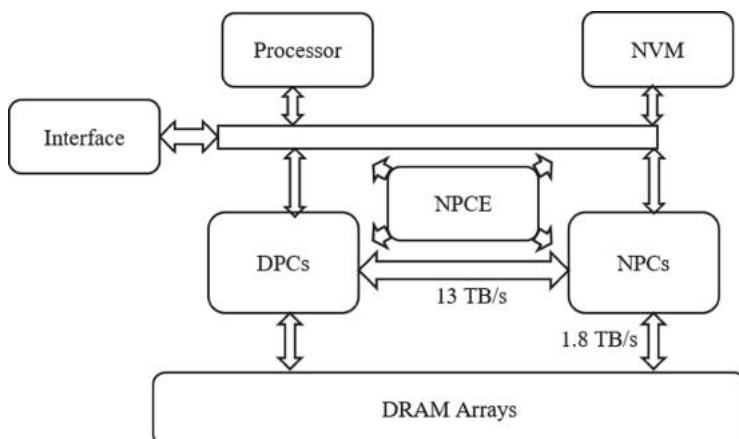


Fig. 6. Chip architecture.

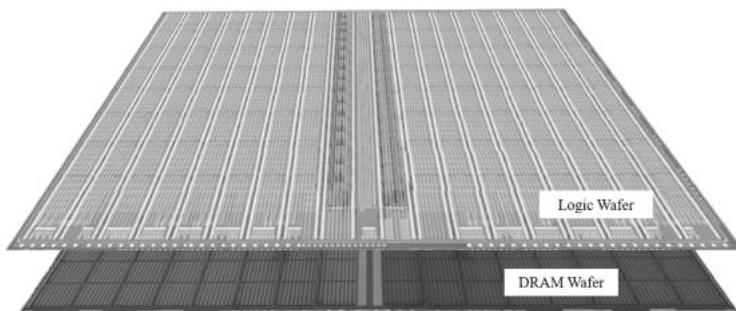


Fig. 7. Wafer-on-wafer stacking.

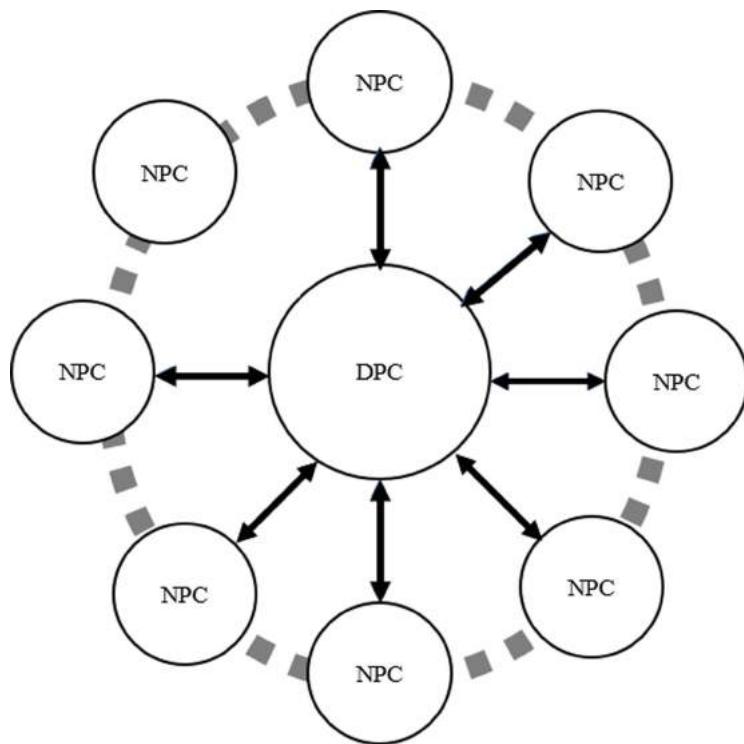


Fig. 8. DPC dispatches data to NPCs and collects computational result. Each DPC Serves 16 NPCs.

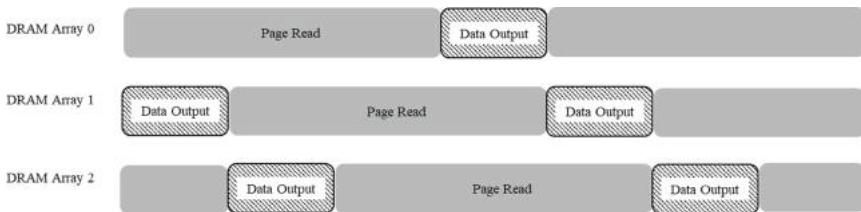


Fig. 9. Time-multiplexed DRAM.

retrieved from DRAM. Thus, APGE reduces the bandwidth and energy requirements for data transfer.

During computation, data is retrieved directly from DRAM arrays. It is then fed into distributed computational pipelines located in the computation cores. The outputs of each pipeline are stored back into DRAM arrays after data rearrangement.

The network processing control engine (NPCE) controls the computation pipelines, as well as the data flow to and from the logic die. While data accesses

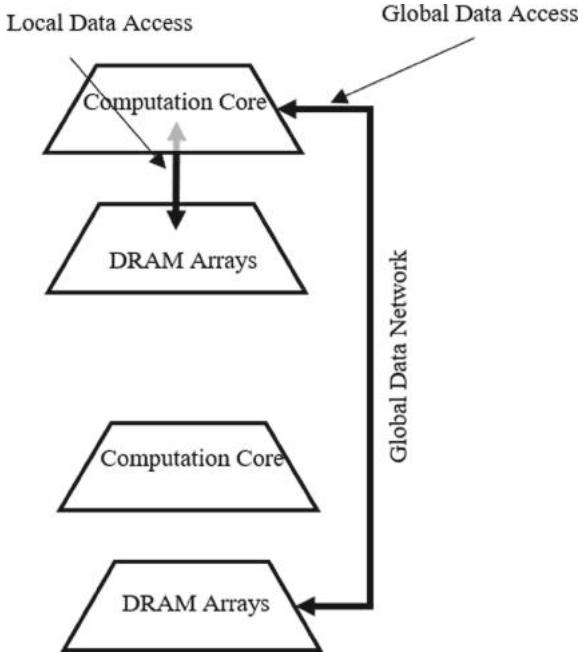


Fig. 10. Data access modes. A computation core can directly access the local DRAM or use the global data network to access further DRAM arrays.

and computation are distributed among the computation cores, all control signals and address for computational units and DRAM arrays are centrally generated by the NPCE. This ensures concordant operation between the two dies. The NPCE consists of a central module and one local module per computation core. The central module manages the above. It runs on a combination of firmware and configuration parameters. The local NPCE modules decode real-time signals from the central NPCE that dictate the computation pipeline and DRAM interface.

NPCE may change computation pipelines dynamically, using switches that enable or disable computation operations (Fig. 11). Real-time pipeline configuration is especially useful for large neural networks that cannot fit onto a single chip. In that case, the chip breaks the network into multiple parts and computes one portion at a time. Each part needs a new configuration. To reduce “dead cycles” during computation and ensure high performance, computation pipelines may change on-the-fly.

Every clock cycle is under direct control of NPCE. As a result, we achieve high utilization of computation units. For example, utilization for RESNET50 inference is 82% (Fig. 14).

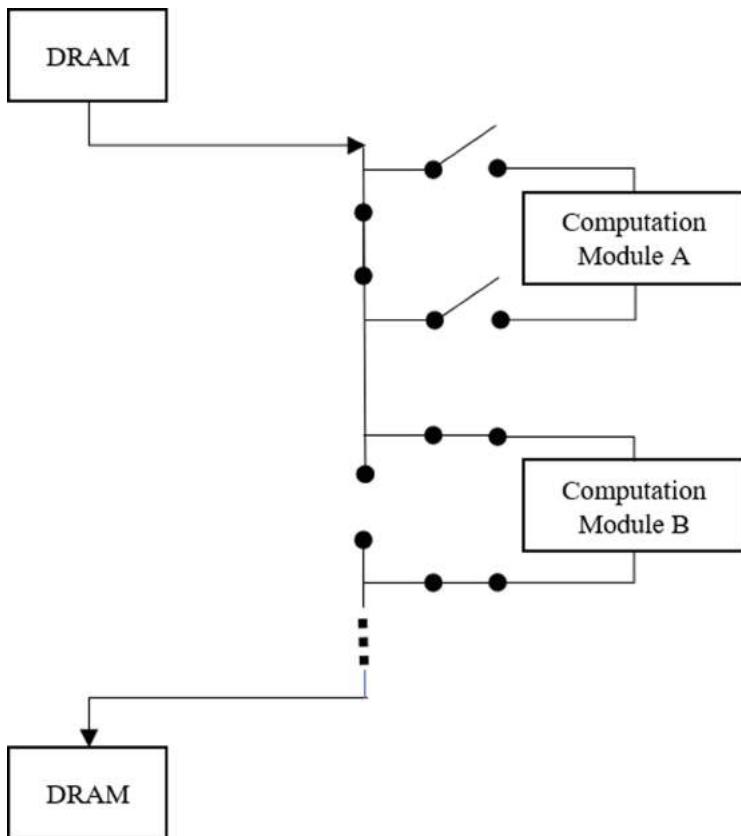


Fig. 11. Computation pipelines can be configured in real-time through switches that enable and disable computation modules.

4 WAFER-ON-WAFER Stacking

DRAM arrays are fabricated with DRAM process on a wafer. The DRAM wafer is packed with DRAM memory cells, data-sensing circuits, and decoding circuits. The logic wafer is fabricated with 40nm technology. DRAM PHY is put on logic wafer. Two wafers are fabricated individually and bonded face-to-face. Electrical connections are made through copper hybrid bonding between aligned metal pad from both wafers (Fig. 13). For chip I/O, through-silicon via on ASIC chips bring I/O to the backside of ASIC chip. Chip I/O pads are made on the backside of ASIC chips and connects through TSV. Process flow for wafer-on-wafer bonding includes two major steps of hybrid bonding and TSV processing (Fig. 12).

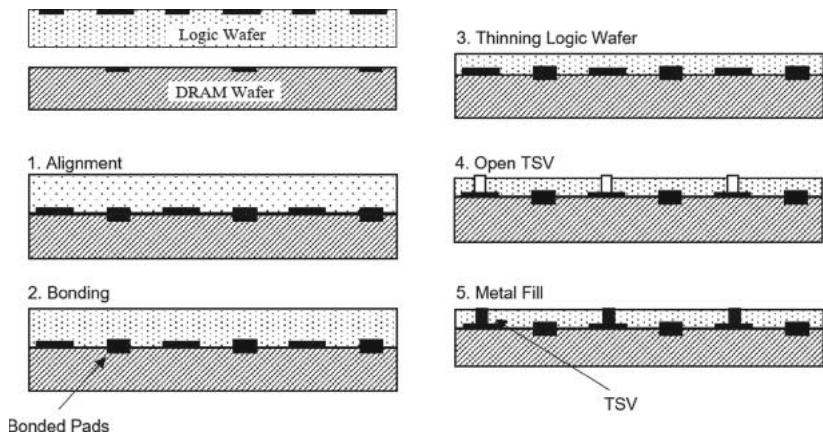


Fig. 12. Process flow for wafer-to-wafer bonding.

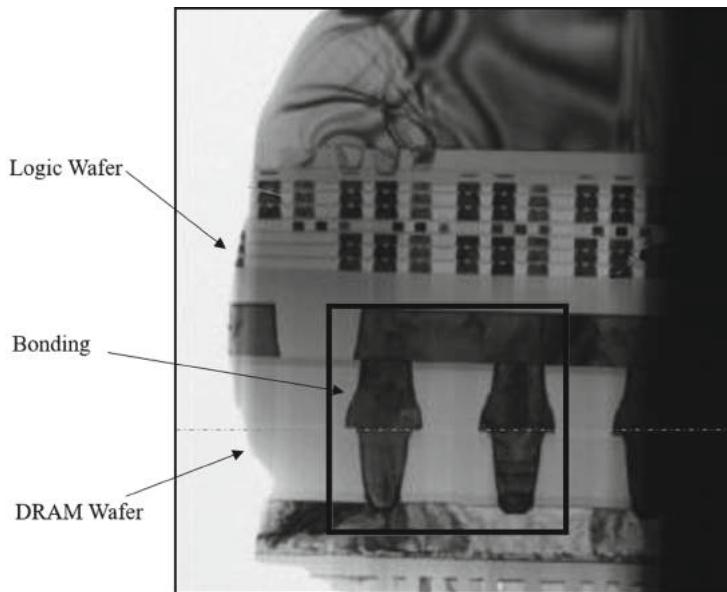


Fig. 13. Cross section of bondings.

5 Results

We fabricate an AI chip specifically designed for the modern demands of deep neural network processing. The chip has two dies bonded. The logic wafer is fabricated with 40 nm process. The DRAM wafer is fabricated with 38 nm DRAM process. Die size is 110 mm^2 ($12.4 \text{ mm} \times 8.8 \text{ mm}$). It has 128 network processing cores 32,768 MAC (multiplier accumulator). Memory bandwidth between

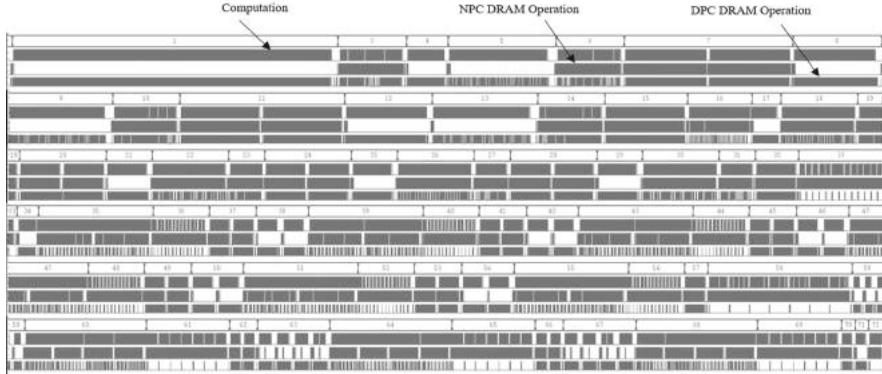


Fig. 14. Concurrent computation and DRAM access for RESNET50.

Table 1. Die-to-die benchmark comparisons

	Peak Performance ^a	Memory Capacity ^b	Energy Efficiency ^c
SUNRISE (40 nm)	0.23	5.11	2.08
Chip A (16 nm)	0.15	0.38	1.02
Chip B (12 nm)	0.18	0.27	0.45
Chip C (7 nm)	1.12	0.07	1.46

^aMeasured in TOPS/mm²

^bMeasured in MB/mm²

^cMeasured in TOPS/W

DRAM chip and logic chip is 1.8 TB/s. Internal memory capacity is 4.8 Gb. Peak performance is 25 TOPS. Typical power consumption is 12 W. Using a combination of network configuration and firmware, data processing on the chip can be customized to any neural network architecture.

Each chip has a different die size. We remove this factor by normalizing by die size and compare the chips on the following peak performance per unit area, memory capacity per unit area and performance per Watts (Table 1):

Sunrise chip outperforms on all metrics except lower performance compared to a chip fabricated at 7 nm. This is understandable considering that Sunrise is fabricated on 40 nm process, a process four generations behind.

6 Conclusion

We fabricate a deep neural network processor chip with wafer-on-wafer bonding. We develop a special architecture that uses DRAM on-chip and removes SRAM cache from the memory hierarchy. The chip delivers high performance and high memory capacity. The techniques for eliminating cache memory are also applicable to a wide range of ASIC devices beyond AI applications.

References

1. Lapedus, M.: Memory Issues for AI Edge Chips, 23 March 2020. <https://semiengineering.com/memory-issues-for-ai-edge-chips/>
2. Khan, S.M., Mann, A.: AI Chips: What They Are and Why They Matter, April 2020. <https://cset.georgetown.edu/research/why-ai-chips-matter>
3. Joel, H.: How L1 and L2 CPU Caches Work, and Why They're an Essential Part of Modern Chips. Extreme Tech, 14 April 2020
4. Microsoft, Turing-NLG: A 17-billion-parameter language model by Microsoft. <https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/>



Study of Residual Networks for Image Recognition

Mohammad Sadegh Ebrahimi^(✉) and Hossein Karkeh Abadi

Stanford University, Stanford, CA 94305, USA
sadegh@stanford.edu

Abstract. Deep neural networks have demonstrated a high potential on image classification tasks while presenting new computational challenges to the machine learning community. Due to the complexity and vanishing gradient problem, it normally takes longer time and more computational power to train deeper neural networks. To address some of these issues, deep Residual Networks (ResNets) can expedite the training process and attain more accuracy compared to their equivalent neural networks without the residual connections. ResNets often achieve this improvement by adding a simple skip connection parallel to convolutional layers in neural networks. Although over the past few years, ResNets have proven to be effective in advancing the performance of deep learning models, the best practices and trade-offs regarding adding residual connections to deep networks, and the exact improvement and disadvantages of these connections during the learning process are not well understood. In this project, we designed ResNet models that can perform a simple image classification task on the Tiny ImageNet datasets. For control, we then compare the performance of these ResNet models with their equivalent Convolutional Network (ConvNet) by removing the residual connections. Our findings illustrate that despite their higher accuracy, ResNets are more prone to overfitting, and that may depend on the depth of the network. We show that several methods to prevent overfittings, such as adding dropout layers and stochastic augmentation of the training dataset can be effective in attenuating this problem in ResNets.

Keywords: Deep learning · Residual networks · Computer vision

1 Introduction

In recent years deep convolutional neural networks have achieved a series of breakthroughs in the field of image classifications [1–3]. Inspired by simple cells and receptive field discoveries in neuroscience by Hubel and Wiesel [4]. Deep convolutional neural nets (CNNs) have a layered structure, and each layer is consisted of convolutional filters. By convolving these filters with the input image, feature vectors for the next layer are produced, and through sharing parameters, they can be learned quite easily. Early layers

Special thanks to Prof. Fei-Fei Li, Dr. Andrej Karpathy, and the rest of the staff at Stanford convolutional neural network class (2016) to guide us through this project.

in convolutional neural networks represent low-level local features such as edges and color contrasts, while deeper layers try to capture more complex shapes and are more specific [5]. One can improve the classification performance of CNNs by enriching the diversity and specificity of these convolutional filters by deepening the network [6]. Although deep networks can have better performance in classification most of the times, they are harder to train mainly:

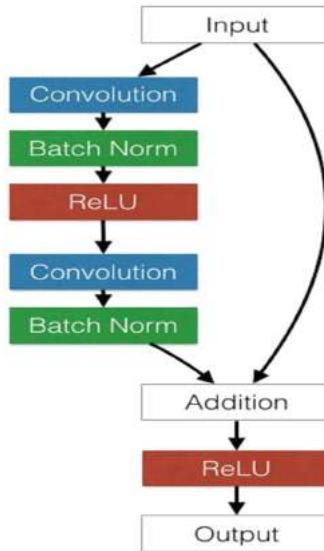


Fig. 1. A RestNet basic block.

Vanishing/exploding gradients: sometimes, a neuron dies during the training process, and depending on its activation function, it might never come back [1, 7]. This problem can be addressed with initialization techniques that try to start the optimization process with an active set of neurons.

Harder optimization: when the model introduces more parameters, it becomes more difficult to train the network. This is not simply an overfitting problem since sometimes adding more layers leads to even more training errors [8].

Therefore, deep CNNs, despite having better classification performance, are harder to train. One effective way to solve these problems as suggested in [9] is Residual Networks (ResNets). The main difference in ResNets is that they have shortcut connections parallel to their normal convolutional layers. Contrary to convolution layers, these shortcut connections are always alive, and the gradients can easily backpropagate through them, which results in faster training. Although deep learning seems very promising in the field of image classification, other methods such as clustering [10] might serve better in the certain applications of online and distributed learning [11]. In this paper, we are going to study ResNets and learn more about the correct ways to use them. In Sect. 2, we explain the different ways are to design a ResNets based on previous works. In Sect. 3, we will describe the tiny ImageNet dataset and Torch, the framework we used for our

implementations. In Sect. 4, we explain the methods we used to design our networks, the basic block that we employed in all our networks, and the stochastic data augmentation technique we used to prevent overfitting. In Sect. 5, we will discuss our results and show how a ResNet compares to its equivalent ConvNet. In our conclusion, in Sect. 6, we will point out a few considerations that must be accounted when designing a ResNet.

2 Related Work

There is a simple difference between ResNets and normal ConvNets. The goal is to provide a clear path for gradients to backpropagate to early layers of the network. This makes the learning process faster by avoiding the vanishing gradient problem or dead Neurons. In the main ResNet paper [9], authors have suggested different configurations of ResNets with 18, 34, 50, 101, and 152 layers. One could describe ResNets as multiple basic blocks that are serially connected to each other, and there are also shortcut connections parallel to each basic block, and it gets added to its output. Figure 1 shows a basic block introduced in [9]. If the input and output size for a basic block are equal, the shortcut connection is simply an identity matrix. Otherwise, one can use average pooling (for reduction) and zero padding (for enlargement) to adjust the size. The author in [12] has compared different basic blocks for one shortcut connection in ResNets (Fig. 1) and shows that adding a parameterized layer after addition can undermine ResNet advantages since there is no fast way for gradients to back through propagate anymore. But considering that condition, there is not a huge advantage or disadvantage for adding an unparameterized layer like ReLU or dropout after the addition module.



Fig. 2. Few sample images from tiny ImageNet datasets.

3 Dataset and Implementation

In this section, we describe the dataset we worked on and the framework we used for network implementations and model training.

3.1 Tiny ImageNet Dataset

In this project we worked on the tiny ImageNet dataset. This dataset consists of a training set of 100,000 images, a validation set of 10,000 images, and a test set of 10,000 images from 200 different classes of objects. All images in tiny ImageNet are 64x64 and so four times smaller than images in the original ImageNet dataset, which have a size of 256x256. Figure 2 shows a few sample images from different classes of tiny ImageNet datasets.

3.2 PyTorch Implementation

Torch is a scientific open-source computing framework with wide support for neural network implementations. In this project, we used this framework to implement and train different ResNet and ConvNet Models. Torch has many predefined neural network layers and also packages that enable us to run our training algorithms on GPUs. We ran all of our models on Amazon AWS GPU nodes.

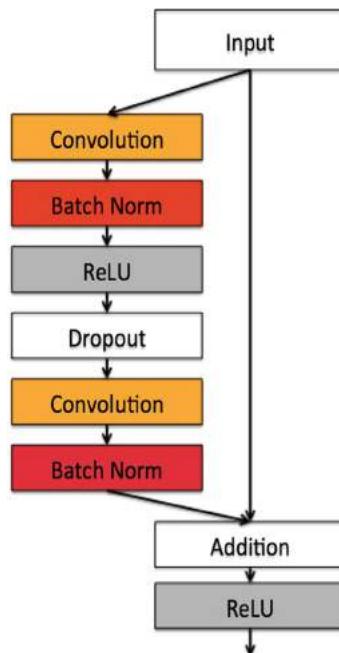


Fig. 3. New basic block with a dropout layer to reduce overfitting.

4 Network Design

The ResNet model introduced in [9] is our starting point for the network design. This model is specifically designed for images in ImageNet and accepts images with size

256 256 and classifies them into 1000 categories. There are many different methods one can employ to start with this trained model and alter it to accept tiny ImageNet images with size 64 64 and classify them into 200 categories. A nave method could be just up-sampling a 64 64 image to a 256 256 and then give it to the trained model, or just skipping the first layer and insert the original image as the input of the second convolutional layer, and then fine-tuning a few of the last layers to get higher accuracy. However, since we're interested in comparing ResNet models with their equivalent ConvNets in this project, we had to design and train our models from scratch (although we might get worse accuracies because of lack of computational resources). In this section, we first describe different network architectures we designed for image classification tasks, and then we illustrate the stochastic data augmentation we used to prevent the model from overfitting.

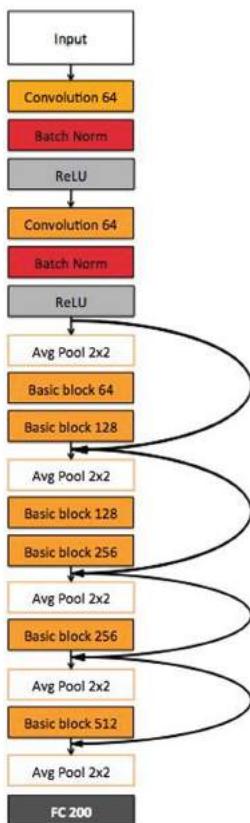


Fig. 4. Example ResNet model for image classification.

4.1 Network Architectures

If we train the original 18-layer ResNet introduced in [9] on a tiny ImageNet dataset, we will see that this model suffers from overfitting. In order to reduce overfitting, we

introduced a new Basic Block (BB) shown in Fig. 3 by adding a dropout layer with parameter 0:5 between the two convolution layers in the basic block shown in Fig. 1. We used ReLU for the nonlinearity unit in all the neurons. Figure 4 shows one of the ResNets we designed for the image classification tasks. This model gives a Top-1 classification accuracy of 49% on the validation set of tiny ImageNet. For more details, see Sect. 5.

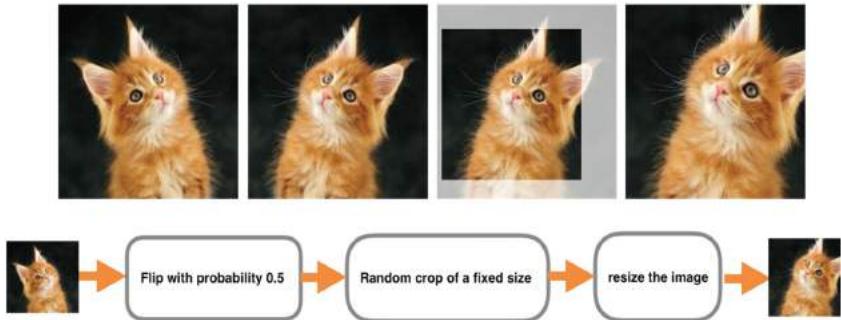


Fig. 5. Online data augmentation of a sample image.

4.2 Data Augmentation

There are only 500 images per each class in tiny ImageNet dataset. This makes the tiny ImageNet to be considered as a small dataset if we're training deep neural networks on them. To overcome overfitting, we need an augmentation method to increase the size of the dataset. One method is to add a few cropped versions of each image and their horizontal flipped images to the dataset. However, since there is limited memory space available, now we cannot load all the images in the new richer dataset into the memory. So, instead of doing all the data augmentations offline, in our implementations, we used an online version of it. Whenever a new batch arrives, we pass all images in that batch from a random transformation unit. This unit first flips the image horizontally with probability 0:5, and then with some probability p , it randomly crops the image to a 56 56 image and then rescales it to its original size, 64 64. Figure 5 shows how this unit works on the sample image. We used $p = 0:7$ in our implementation.

5 Results

As mentioned before, even though the vanishing gradient problem is a big issue for deep neural networks, in shallow ConvNets, it is not a big deal. In order to observe this effect, we compared two shallow networks with 7 and 9 layers. Figure 6 and 7 show the loss function and training and validation accuracy of these two networks on the CIFAR-10 dataset. As we see, for a 9-layer network, ResNet and ConvNet have similar performance, and for even shallower networks (7-layer), the ResNet performance is even worse than plain ConvNets. This result makes sense because when you are adding the output of a

convolutional layer with its input, you are basically averaging a trained processed data with the raw data, and that would just harm the training if there were no other benefit to it. But if there are other benefits to it (for example, in deep networks), the overall effects could be improved accuracy.

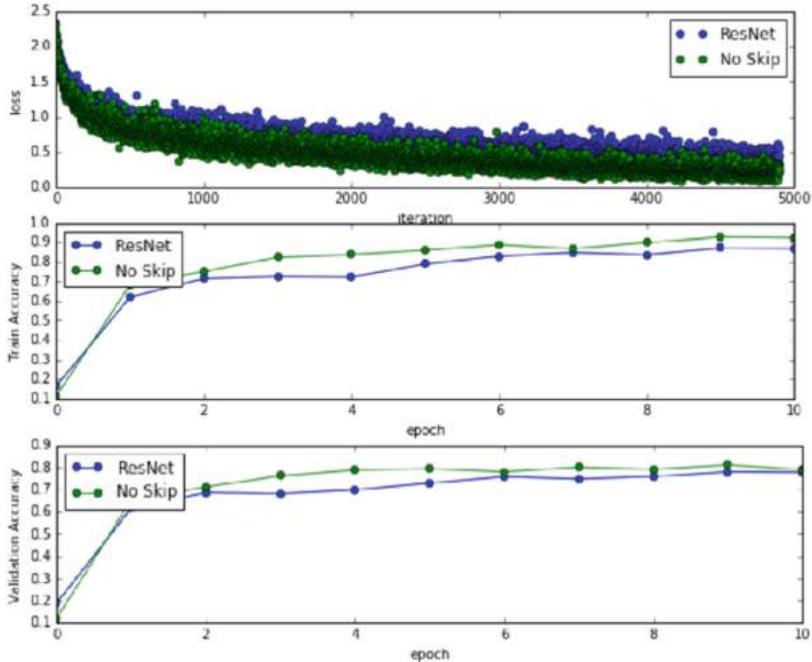


Fig. 6. Training/validation accuracy and loss at each epoch for a 7-layer network over the CIFAR-10 dataset.

Then we tried to train multiple deep ResNets varying from 12 to 21 layers and see which one performs better on the Tiny ImageNet data set. The results are brought in Table 1. Note that all the models are trained for the same number of epochs. We picked the best performing network (Net 1) with 49% percent validation accuracy and trained an equivalent plain ConvNet with the same architecture (Net 6). In Fig. 3, we compared the accuracy of these two networks (ResNet and equivalent ConvNet). One could clearly see that the ResNet has much higher accuracies than plain ConvNet, and it trains much faster. In this ResNet, the validation accuracy, 13%, and training accuracy 30% percent higher than its ConvNet equivalent. The difference between training accuracy and validation accuracy is a good indicator of overfitting, and based on our results, we realized that ResNets are more prone to overfitting. In Fig. 8, one can see that this difference for plain ConvNet is 7%, while in ResNet, it is around 23%. Figure 9 shows how loss decreased while training both models. Originally this difference was even higher for ResNet (around 30%), but we used the dropout and stochastic augmentation technique that was described in Sect. 3 to reduce this overfitting but could only reduce it by 6%. Another way to reduce the overfitting is to have a smaller parameter set, which means

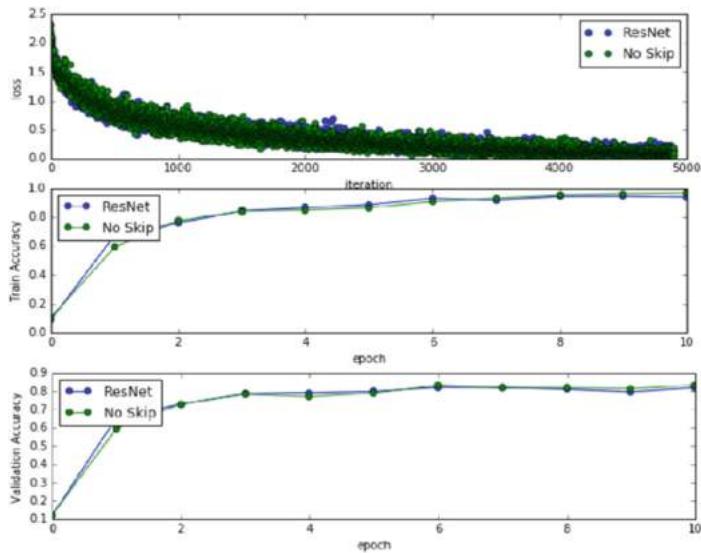


Fig. 7. Training/validation accuracy and loss at each epoch for a 9-layer network over the CIFAR-10 dataset.

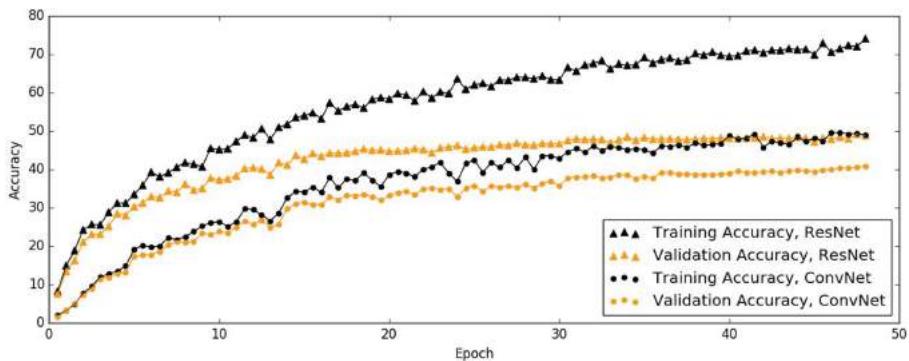


Fig. 8. Training and validation accuracies for ResNet Net1 are described in Table 1 and its equivalent ConvNet.

fewer convolution layers. In Net 3 (Table 1), we implemented such a network, and the training and validation accuracy difference was reduced to 16%, but the downside of this model was to have smaller validation accuracy in comparison to the best network (by 3%). Both dropout and stochastic augmentation was used in this implementation.

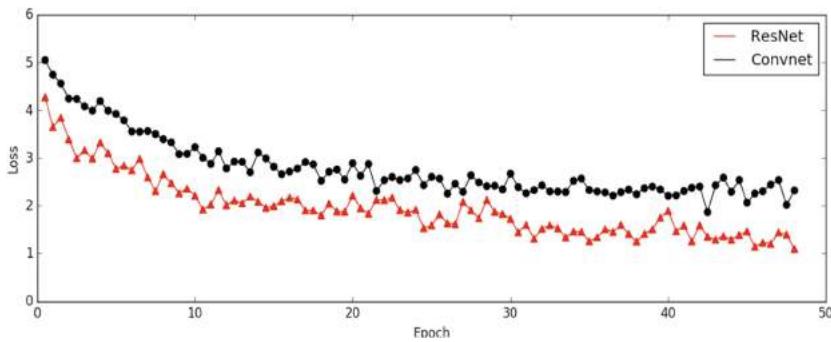


Fig. 9. Loss vs. epoch in the training of the ResNet Net1 described in Table 1 and its equivalent ConvNet.

Table 1. Training and validation accuracies of different models.

Net 1	Net 2	Net 3	Net 4	Net 5	Net 6
(Conv 64) 2	Conv 64	Conv 32	(Conv 64) 2	(Conv 64) 2	(Conv 64) 2
Avg 2	Avg 2	Conv 64	Avg 2	Avg 2	Avg 2
BB 64	BB 128	Max 2	(BB 64) 3	BB 128	(Conv 64) 2
BB 128	Avg 2	(BB 64) 2	Max 2	Max 2	(Conv 128) 2
Avg 2	BB 128	Avg 2	(BB 128) 3	(BB 256) 2	Avg 2
BB 128	BB 256	(BB 128) 3	Max 2	Max 2	(Conv 128) 2
BB 256	Avg 2	Avg 2	BB 256	(BB 512) 2	(Conv 256) 2
Avg 2	BB 256	BB 128	BB512	Avg 2	Avg 2
BB 256	Avg 2	Avg 2	Avg 2	Dropout	(Conv 256) 2
Avg 2	BB 512	BB 256	Dropout	FC 200	Avg 2
BB 512	Avg 4	Avg 2	FC 200		(Conv 512) 2
Avg 2	FC 200	FC 200			Avg2
FC 200					FC 200
15	12	17	21	15	15
72%	69%	62%	44%	55%	43%
49%	46%	46%	36%	42%	36%

6 Conclusion

As we explained in our results, adding a simple shortcut connection can improve the accuracy in the image classification task and make the training process much faster. But the trade-off is that residual networks are more prone to overfitting, which is undesirable. We showed that by using different machine learning techniques like drop out layer and stochastic augmentation, we can reduce this overfitting, and if designed properly, we can

have fewer parameters that result in much smaller overfitting (14%). We also observed that ResNets are more powerful for very deep networks, and if employed improperly, it could even hurt the performance for very shallow networks. To conclude, ResNets show a promising landscape in deep learning, but it should not be just blindly used, and there is a lot of room to study and understand their functionality and correct use.

References

1. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (2010)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Commun. ACM* **60**(6), 84–90 (2017)
3. Rolet, P., Sebag, M., Teytaud, O.: Integrated recognition, localization and detection using convolutional networks. In: Proceedings of the ECML Conference (2012)
4. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* **195**(1), 215–243 (1968)
5. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) *ECCV 2014. LNCS*, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
6. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint [arXiv:1409.1556](https://arxiv.org/abs/1409.1556) (2014)
7. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5**(2), 157–166 (1994)
8. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. arXiv preprint [arXiv:1505.00387](https://arxiv.org/abs/1505.00387) (2015)
9. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. Li, S., Kar, P.: Context-aware bandits. arXiv preprint [arXiv:1510.03164](https://arxiv.org/abs/1510.03164) (2015)
11. Mahadik, K., et al.: Fast distributed bandits for online recommendation systems. In: Proceedings of the 34th ACM International Conference on Supercomputing (2020)
12. Sam Gross, M.W.: Training and investigating Residual Nets (2016)



A Systematic Review of Educational Data Mining

FangYao Xu¹, ZhiQiang Li², JiaQi Yue³, and ShaoJie Qu²(✉)

¹ School of Mathematics and Statistics, Beijing Institute of Technology,
Beijing, China

1120180084@bit.edu.cn

² Network and Information Center, Beijing Institute of Technology, Beijing, China
{lizq, qushaojie}@bit.edu.cn

³ School of Computer Science, Beijing Institute of Technology, Beijing, China
3220201000@bit.edu.cn

Abstract. As an important part of data mining, educational data mining (EDM) has played a significant role in the field of education. This article reviews the development process of EDM, summarizes limited research directions of EDM from the past five years, and analyses the motivation and purpose of these studies, including the problems that have been solved. Next, the tools and algorithms used are introduced according to the referenced literature. Finally, this article discusses and summarizes further research directions based on previous research results and some thoughts with the hopes of offering researchers future research directions.

Keywords: Improving classroom teaching · Pedagogical issues · Media in education

1 Introduction

The promotion of new teaching methods and teaching platforms enables the timely retention of various data generated by students during the learning process. These data contain the basic background data of students, the grades of the students during school, the information of the learning website browsed, the text information of the course feedback, etc. To make good use of the available information that reflects students' learning status and learning process, data mining is needed to extract useful instructive or predictive information [60]. This information provides a basis either to predict changes in student performance, to provide early warning for potential poor students, or provide new feedback for teachers to improve current teaching methods [75]. Hence, more research experiments have been performed to explore and research [24].

Many new learning platforms, such as the famous massive open online course (MOOC) learning platform, have begun to emerge. The data generated by students in the learning process are no longer limited to digital data such as grades

and attendance rates. An increasing number of data forms that store student data have become the object of research, such as text, web browsing, blog and image data. These data contain a large amount of information about students' learning, and the patterns revealed by educational data mining (EDM) technology play a large role in improving teaching quality and students' performance in the actual education system. The most crucial task is predicting students' performance. To forecast students' grade point averages(GPAs), many factors need to be considered, such as income, and previous academic performance by using EDM [29]. In school, negative or positive feedback from students means much to instructors and applying EDM to analyse this feedback could help teachers improve better to some degree [80]. EDM can provide feedback to users by finding the patterns in the data collected in a new environment, to provide useful information for students, teachers and decision-makers to facilitate improvement. However, as a relatively new field, there are still many gaps in educational data mining. Actually various student learning data collected from learning platforms present graph data structures, which makes machine learning algorithms unable to handle this well. Hence, how to use graph neural network in educational data mining to improve teaching effect is worthy of attention, such as presenting graph data structures and selecting more factories which affect students' academic performance.

The rest of this paper is structured as follows: Sect. 2 demonstrates some related work. Section 3, mainly focuses on reviewing the research directions, and analysing the motivation behind them. Next, in Sect. 4, we roughly divide the related methods into three categories and describe them. Then, we provide some discussion and a limited list of research directions and holes, hoping to inspire future researchers. Finally, Sect. 6 draws conclusions from the whole paper.

2 Related Work

To manage and use various information, designing data warehouses can contribute to extracting, transferring and loading educational data to further analyse educational data better [63]. Organizing knowledge is necessary for web-based learning systems. Thus, a concept map was proposed [3]. Data visualization has been highly emphasized, and it has been indicated to be useful and some interesting hidden relationships in the numbers have been revealed [50]. In the real world, some of the collected data can be incomplete, erroneous or missing, and the imbalanced distribution of data has a large impact on the effect of some algorithms. Therefore, scholars have proposed a new algorithm in the light of density-based approach, to address incomplete data clustering tasks [19].

To assess student performance and predict whether they drop out of school, scholars have used Pearson product moment correlation (PPMC) to reveal some hidden relationships behind different variables, and the J48 algorithm to predict whether students fail or succeed in academia and utilized multiple regression to predict GPA [38]. Some researchers used data based on a small private online

course (SPOC) and a gradient boosting decision tree (GBDT) to build a predictive model to determine who is at risk; these researchers concluded from their experiment that this model performed well [30].

Predicting the dropout rate has always been a concern. Taking into consideration factors such as the total number of employees, the total number of computers, and the total number of rooms, some scholars adapted two non-parametric techniques to predict student dropout [67]. By comparison, the results showed that support vector regression outperformed non-parametric quantile regression and some improvement measures were provided.

To evaluate the quality of the teachers' courses, some studies have used students' feedback on course questionnaires and machine learning algorithms to provide teachers with guidelines [4]. Another study focused on the user experience of the teacher's establishment of a predictive student performance model, which could provide teachers with some guidance for classroom improvement and improve teaching effects [83].

As a new teaching form, audio teaching has attracted the attention of some scholars. For such a new research field, only a few scholars have reviewed EDM technologies and provided reference bases for audio educators [71]. More research on this topic is needed, this topic will be introduced in more detail following the next chapters.

3 Research Directions

3.1 Traditional Teaching

Learning Analytics. Learning analytics (LA) is the measurement, collection, analysis and reporting of data about learners and their contexts, for the purposes of understanding and optimizing learning and the environments in which it occurs [41]. Some scholars have aimed to review the research prospects of LA to understand the research activities in this multidisciplinary field [86]. However, LA, consequently, raised serious issues concerning student privacy, autonomy, and the appropriate flow mechanisms of student data [41]. Some relevant rules were proposed to ensure that educational data is used by and stored at other institutions legally while considering the security of students' private information [10]. Some studies discussed the relationship between LA and students' information privacy, and proposed some research directions [28].

Sentiment Analysis. Sentiment analysis is an extremely important cross-cutting method. By performing a sentiment analysis on students' feedback on a course, it can be determined whether the students liked the course and some possible ways to improve the course can be suggested to the teachers. Sentiment analysis addresses the feedback of students, to obtain the degree of students' love of the course, which in turn can show teachers how to adjust the content and form of the classroom. There has been research on the link between emotional

analysis and performance [16]. The results showed that feedback was more effective when emotional content was not included, which is contradictory to common sense; therefore, this is still an unresolved problem. To analyse feedback from a teaching evaluation questionnaire, and aid in the selection of excellent teaching staff, the authors used text sentiment analysis to quantify students' textual opinions, and used students' sentimental tendencies to comment on faculty [84]. An outstanding problem of online platforms is that the registration rate of courses is very high, but relatively few people actually complete them. Based on the above demand, researchers [89] proposed a semantic analysis model to track learners' emotional tendencies, and analyse the acceptance of a MOOC based on big data from homework and forums. Some research has used big data frameworks to simulate and predict student emotions (amusement, anxiety, boredom, confusion, excitement, excitement, depression, etc.), which has made substantial progress [40].

Prediction of Poor Students and Accessing Students' Performance. In universities, students are very likely to fail some course or even drop out. Therefore, evaluating the performance of students, and predicting students who may have poor academic performance, are important issues in EDM. Researchers have tried to judge students' performance through attendance, unit tests and other variables, and concluded that their performance depended on unit tests, and attendance, and certain measures could be taken to help students who performed poorly on unit tests [14]. In terms of accessing performance, scholars studied the learning behaviour of students through network teaching platform data through association rule mining [9]. Author [46] focused on selecting attributes with high influence on students' performance in Indonesia. Some scholars did not fully use educational data, and considered factors such as parents' occupations to predict student performance [76]. The above research also pointed out a new research direction, that is, how to use less educational data and more non-educational data to predict student academic performance. A deep learning(DL) algorithm was proven to be excellent and better than conventional methods [47]. A neuro-fuzzy system combined with classification was proposed to predict the academic achievement of engineering students [15]. The ordered decision tree algorithm using ordered entropy was applied to the comprehensive quality evaluation of college students [31]. More DL algorithms could be investigated in the prediction direction.

Student Dropout Prediction. After assessing student performance and determining students who are at risk of dropping out, interventions can be implemented. However, it is still of great importance to predict the dropout rate. Table 1 introduces limited tasks and notes the contributions of each work.

There are many reasons why students lose the motivation to work hard in university, such as too much temptation or courses that are too difficult, which can easily cause students to drop out. How to use useful information provided by EDM to reduce the dropout rate is particularly important. In Brazil, urgent

Table 1. Solved problems and contributions of various works

Reference	Task(s)	Algorithms	Tools	Contributions
[25]	Compare open source tools	K-means	Weka, Knime Rapid Miner	Provided guidance for identifying students needing attention
[39]	Discuss various prediction techniques	J48, SVM, Naïve Bayes, etc.	Weka	Multilayer perceptron performed best
[62]	Comparison between different algorithms	Naïve Bayes, J48, random tree, etc.	Weka, Orange3, Rapid Miner	J48 and random tree performed better
[32]	Predict student dropout	Naïve Bayes, decision tree	R language	Prevented students from dropping out and provided counseling
[6]	Propose new prediction model	SVM, decision tree, etc.	None	Proved ensemble strategy is efficient
[77]	Identify profile of at-risk university students	NN, SVM, naïve Bayes, random forest, etc.	scikit-learn library	Found potential dropout patterns
[59]	Explore how to discover pedagogically relevant knowledge	K-means classification, association rules	Excel, clementine, SQL, etc.	Provided support to students and teachers
[2]	Propose prediction model, examine ML applications	Logistic regression, SVM, naïve Bayes	Excel	Clarified effective features and methods

solutions for the high dropout rate are required. Due to the above demand, taking into consideration the number of courses approved and so on, scholars have put forward some suggestions to improve the local educational situation [56]. DL algorithms have also been used in prewarning mechanisms [27], and the results were excellent. In addition, novel ideas have been proposed to predict student drop out. A method based on thresholding outperformed existing approaches

after being tested on two datasets to predict drop out [82]. Some scholars established time series classification models to predict the learning status of students [91]. In reality, addressing this problem without using existing algorithms is a problem that can be considered.

Decision-Making Support. It is necessary to obtain useful indicative information by EDM to support decision-makers in making appropriate decisions to improve teaching. A model based on a specific neural network(NN) combined with other data mining approaches increased the effectiveness of the decision-making process at any time and supported automatic knowledge acquisition [48]. DL is a subfield of machine learning that uses NN architecture to model high-level abstractions in data [33]. Moreover, business intelligence and tools applied in companies can also make great contributions to decision-making in educational institutions, and have been shown to be feasible and manageable [85]. Researchers have used fuzzy representation technology to inductively investigate questionnaire data and proposed a tool that can display results, information, explanations, comments and suggestions to non-expert users of data mining in a meaningful way [9].

Improving Teaching Effects. By mining the patterns in data, the information obtained can improve teaching effectiveness. Association rules are used to explore different elements and certain underlying relationships with users [9]. Researchers have analysed the correlation between online learning attributes and academic achievement, explained that good online learning habits were associated with excellent performance and proposed appropriate methods to improve teaching effects through cluster analysis [21]. In addition, some researchers have suggested way to enhance the use of learning course management system and encourage students to be more active in blended learning models [11]. Additionally, in order to understand how teachers reflect in professional learning environments, some scholars used inductive content analysis and single-label text classification algorithms to analyse teachers' online discussion text data [52]. This study proposed a recurrent NN (RNN)-based teacher evaluation comment mining model, which can help teachers improve teaching effects by providing feedback [68].

Reviewing and Providing Insight into EDM. Some scholars have aimed to provide a perspective of EDM [75]. Moreover, others have comprehensively introduced various EDM techniques and data sets used from a variety of modern education models to provide overall insight into EDM for researchers and non-professionals [74]. Next, some research has also provided a review of present software, which can greatly reduce the time spent in data mining [48]. Then, some scholars tried to describe tools and methods of both data mining and EDM [62]. This research [33] provided us with a comprehensive perspective to understand DL, such as basic concepts and several important algorithms. In addition, the detailed procedure and tools of classification were also discussed [15]. Additionally, some scholars have provided a systematic review of the educational text

mining, including analysing different text processing methods and data formats [26].

3.2 Distance Education and Online Learning

Educational process mining and audio teaching are new problems that have emerged in only the past two years. The emergence of the MOOC platform promotes the following two sections of research. Often, it is difficult to implement the teaching process according to the specific personality of the students through the MOOC platform. The author in [49] proposed a hybrid NN model to predict students' learning methods. By determining the learning methods, learners can effectively improve their learning efficiency. Some lecturers monitor student participation and analyse its impact on students' performance to enhance course quality. Therefore, researchers have proposed a semantic network model for measuring different word associations between teachers and students to measure students' participation in MOOCs [51].

Educational Processing Mining. Educational processing mining (EPM) uses log data specifically collected from the educational environment to discover, analyse and offer a visual representation of the complete educational process. The author in [13] introduced EPM and detailed the potential of this technology. It also described other related fields, such as intentional mining. Some researchers proposed a method based on process mining to evaluate the performance of students performing certain tasks on a computer [12]. Considering MOOCs, to predict student performance, this study analysed the aggregated activity frequency, frequency of specific course items, and activity sequence by using correlation, multiple regression and process mining [22]. As an emerging field, there are many research directions that can be discussed in depth.

Audio MOOC Related Questions. The phonetic learning system is a powerful learning tool for illiterate and semi-illiterate individuals, but related evaluation and introductory research are lacking. Some researchers have conducted a comprehensive evaluation of the existing audio learning system and have made great contributions to the research in this area [61]. Some scholars have performed research on predicting the popularity of video courses and proposed a method with a good approximation ability, providing video producers with some suggestions for improving courses [54]. However, related research is still lacking, and more in-depth research is needed.

4 Methodology

4.1 Machine Learning

With continuous development of educational data mining, more and more algorithms are applied to preprocess data, analyse data and draw useful conclusion

Table 2. A brief comparison of the advantages and disadvantages of five clustering algorithms

Serial number	Algorithm	Advantages and disadvantages
1	Mean-shift	Can detect the center-point
2	K-means clustering	Well-behaved, but relies on a prior knowledge of noise
3	K-mediods	Reduces pairwise variations and more robust relatively
4	DBSCAN	Arbitrary shape clusters insensitive to outliers
5	Hierarchical clustering tree	Can select the best cluster

for educational purpose, including machine learning, neural networks and graph neural networks. Owing to the variety of algorithms, it is difficult to decide which algorithm(s) to use in different situations, so some scholars have tried to give a comprehensive introduction. Some scholars have performed relevant research and gave the results under specific datasets [59]. As an important means of data mining, machine learning-related algorithms naturally play a very important role in EDM to predict and evaluate [2,44]. Researchers tested four machine learning algorithms to decide which methods and which features worked best with the data in a specific dataset [59,79]. However, some scholars discussed the advantages and disadvantages of machine learning algorithms [15].

Cluster analysis is a viable data mining method to analyse learning behaviours in online learning environments (OLE) [69]. According to the above mentioned article, the authors conducted an exhaustive review of clustering algorithms based on their motivation, understanding of the output, assessment of the cluster validity and so on to provide instructions for non-professionals. Additionally, some scholars analysed clustering on various aspects related to cluster principles, implementation methods and data processing, and demonstrated the usefulness of clustering through examples [69,70]. Moreover, researchers analysed and discussed the advantages and disadvantages of a few clustering algorithms in Table 2 exhaustively, including mathematical principles [45]. With the help of Weka software, several algorithms, including the J48 algorithm and multilayer perceptron (MLP), were tested based on KALBOARD360 and compared; the results showed that MLP performed best [39]. Some classification methods were introduced, such as distance method, rule-based method and status-based method. Some studies compared Bayesian networks and decision tree [46]. Some scholars compared the effectiveness and accuracy of the random forest model and the J48 classifier in university recruitment [7].

To better predict the performance of students, some studies used feature analysis and machine learning algorithms, and tested different combinations of

machine learning algorithms based on Gini coefficients and p-value [87]. Machine learning algorithms can also be used in text data. It has been proven that excellent results can be achieved in text-based data by Naïve Bayes classifier if the requirements of the Bayes algorithm are met, which means that the features are independent of each other [39]. Some scholars have studied game-based courses that taught EDM techniques related to classification and association rules through different environments in the game [17]. Owing to the need to study the potential of community questions and answer websites, some scholars have used natural language processing methods and machine learning algorithms to classify data [65]. In addition, we also summarize the research directions of some scholars as shown in Table 3.

Table 3. A brief summary of the problems solved using machine learning algorithms

Reference	Task(s)	Algorithm(s)	Dataset
[72]	Compare machine learning algorithms	SVM, J48, etc.	MOOC platform
[73]	Explore imbalanced classes by SMOTE and OSS	SVM, apriori	UCI machine learning repository
[92]	Explore the relation between objectives and student outcomes	Apriori, association rules mining	152 self-study reports
[35]	Explore categories and characteristics of student	Clustering algorithms	Unknown
[57]	Study the potential of adaptive learning platforms	Logistic regression	Osmosis
[66]	Provide decision support for tutoring	K-means clustering etc.	Unknown university

The function of existing algorithms in actual data sets is bounded. The ensemble strategy can overcome some shortcomings of the algorithms. Nevertheless, research in this area is relatively lacking, which calls for exploration by future scholars. There is an area to study the ensemble strategy, which does improve the results of the existing algorithm to a certain extent [1,6].

4.2 Deep Learning

To tackle task of automatic short answer grading (ASAG), deep belief network (DBN) was utilized and compared with other classifiers. The new proposed model

outperformed the conventional model and the discussed classifiers [95]. In fact, the author mentioned that grading student essays is a challenging and complex task, that warrants further study. This paper proposed an emotional education framework based on learner interest and emotion recognition, and used artificial NN with DL (ANN-DL) method to study student's emotion recognition [20]. Some articles discussed how to use recurrent NN architecture in educational research in two small cases, which refers to a two-layer long short-term memory (LSTM) network in this paper [81]. To predict whether students will accept the academic placement proposal, the NN model and other machine learning algorithms were used for comparison, and the results showed that the NN was better [78]. A back-propagation NN was used to mine and classify personnel educational training data of Chinese automobile companies, and a demand model suitable for educational training of other related industries was proposed [34]. In this study [36], a deep ANN was deployed to analyse clickstream data extracted from virtual learning environments to predict at-risk students.

4.3 Other Methods

When making predictions and classifications, not all the labels will play a certain role, so it is important to select meaningful features. Comparison of different feature selection algorithms is essential to understand more about their advantages and disadvantages [93]. The author in [94] compared three feature selection algorithms and used a support vector machine model (SVM) to compare their effectiveness. In addition, it proved that when taking different data pre-processing measures, the effectiveness of models will be affected greatly [8]. Additionally, some scholars proposed a combination of the synthetic minority over-sampling technique (SMOTE) and one sided selection (OSS) for data pre-processing, which surely improved the accuracy of the SVM [73]. To promptly help high-risk students who need to complete their studies, some researchers used students' Internet access logs to predict high-risk students [96]. However, the high-dimensional raw data obtained were complex and noisy. Therefore, several new data pre-processing methods were proposed. As a special kind of data, text information requires unusual processing methods. Due to the importance of vocabulary knowledge, some researchers designed a system that can predict the contextual information of target words in the difficult range from junior high school to university [42]. Scholars have proposed a grammar-based genetic programming method to limit the search space and include grammatical constraints to mine context-aware association rules [53]. The use of discrete-time Markov chain (DTMC) and hidden Markov model (HMM) methods were recommended for analysis and processing to predict student performance in short time series [64]. Some scholars have innovatively used partial least squares structural equation modelling (PLS-SEM) to develop models describing how online behavioural participation affects achievement [88].

5 Discussion and Future Work

5.1 Discussion

Some scholars have conducted relevant research on end-users' acceptance of EDM technology to learn more about students' responses to present educational methods, and platforms and analyse students' performance by EDM [90]. A combination of concept maps and generic methods is worth exploring for particular test records [3]. Moreover, apprentice learner architecture, a computational theory of learning was proposed, and the author explored the inner relationship between learning and educational data [55]. The novel idea was put into operation, and the results are inspiring. In terms of using frequent patterns, a multi-granularity pattern-based sequence classifier was proposed to detect the roles of students working on a project and interacting, and this new approach gave better results than previous methods [37]. Even though the technical methods of EDM are varied, the purpose is to improve the effectiveness of education.

5.2 Future Work

As mentioned above, the research direction of EDM is constantly being broadened. Researchers have also mentioned some general directions of EDM [33]. In this section, we will systematically propose some future research directions. The following are some problems proposed by previous scholars that can be explored in the future:

1. Considering database imbalance, we can improve the present threshold-based method to solve new problems [82]. We can study how to apply SMOTE and OSS to other machine learning algorithms to improve their performance [73].
2. Educational process mining and audio course teaching are both newly emerging fields with great research potential [13, 61].
3. As the author of [55] mentioned, EDM researchers are expected to cooperate to discover the potential of apprentice learner architecture to have a collective understanding of human learning.
4. Some scholars research time between actions (TBA) to explore students' learning patterns, and the results also confirm the researcher's conjecture. As the researcher asserted, further research on this phenomenon is necessary [18].
5. Sentiment analysis in the field of text mining is a hot topic. Textual data require more attention [58], and it is possible that a tool could be designed to integrate and process all existing natural language processing methods [26].
6. It should be determined whether it is possible to design a set of algorithms so that the various data types available can be transformed into suitable forms and applied to experiments [43] and the problem with the data itself can be solved.
7. Data standardization is a cornerstone aiming to resolve the different data formats, so how to standardize data better is of great significance [75]. Further, new methodologies should be put forward to better clean educational data [5].

8. Personalized learning is worthy of paying more attention and it is quite challenging [5, 60]. How EDM is utilized in audiology is also a new field that requires further exploration [71].
9. Few professions pay attention to gamified e-learning systems. This may be a new field to explore to some degree [23].

By reviewing the existing literature, we have raised the following questions and directions that can be further explored:

1. How to apply DL to explore patterns behind textual information or emotional analysis more deeply by EDM should be explored.
2. By mining association rules, too many rules will be revealed ; thus, selecting the most meaningful rules for actual demand is essential.
3. It is worthwhile to explore combinations of different algorithms, including the incorporation of optimization algorithms, to improve the accuracy of the model results.
4. We can try a variety of the algorithm combinations to improve the accuracy of models results. Of course, we can also consider the incorporation of some optimization algorithms.
5. How to reduce the impact of incorrect data and missing values, and new algorithms to impute the missing values are worth considering.

The above list is only a fraction of the problems to be solved, and some are mentioned in the future research direction section. By continuously fostering the development of EDM technology, more fields will benefit from it.

6 Conclusion

The greatest contribution of this article is a systematic and comprehensive review of the research directions, which shows a considerable part of the research directions in the field of EDM. In addition, methodologies and tools are also mentioned and discussed. Finally, we summarize the remaining areas that need to be deepened in the preceding research directions, and propose some ideas. We hope this article can provide guidance directions for non-professional scholars, future reserchers, and scholars who are studying EDM.

Conflicts of Interest. There are no conflicts of interest to declare.

References

1. Abdar, M., Zomorodi-Moghadam, M., Zhou, X.: An ensemble-based decision tree approach for educational data mining, pp. 126–129 (2018)
2. Abe, K.: Data mining and machine learning applications for educational big data in the university. pp. 350–355 (2019)
3. Acharya, A., Sinha, D.: An educational data mining approach to concept map construction for web based learning. *Informatica Economica* **21**(4), 41–58 (2017)

4. Agaoglu, M.: Predicting instructor performance using data mining techniques in higher education. *IEEE Access* **4**, 2379–2387 (2016)
5. Aghabozorgi, S., Mahrooeian, H., Dutt, A., Wah, T.Y., Herawan, T.: An approachable analytical study on big educational data mining, pp. 721–737 (2014)
6. Ajibade, S.S.M., Ahmad, N.B.B., Shamsuddin, S.M.: Educational data mining: enhancement of student performance model using ensemble methods. In: Iop Conference, p. 012061 (2019)
7. Algur, S.P., Bhat, P., Kulkarni, N.: Educational data mining: classification techniques for recruitment analysis. *Int. J. Mod. Educ. Comput. Sci.* **8**(2), 59–65 (2016)
8. Amrieh, E.A., Hamtini, T., Aljarah, I.: Preprocessing and analyzing educational data set using X-API for improving student's performance, pp. 1–5 (2015)
9. Angeli, C., Howard, S.K., Ma, J., Yang, J., Kirschner, P.A.: Data mining in educational technology classroom research: can it make a contribution? *Comput. Educ.* **113**, 226–242 (2017)
10. Askinadze, A., Conrad, S.: Respecting data privacy in educational data mining: an approach to the transparent handling of student data and dealing with the resulting missing value problem, pp. 160–164 (2018)
11. Ayub, M., Toba, H., Wijanto, M.C., Yong, S.: Modelling online assessment in management subjects through educational data mining (2017)
12. Baykasoglu, A., Ozbel, B.K., Dudakli, N., Subulan, K., Senol, M.E.: Process mining based approach to performance evaluation in computer-aided examinations. *Comput. Appl. Eng. Educ.* **26**(5), 1841–1861 (2018)
13. Bogarin, A., Cerezo, R., Romero, C.: A survey on educational process mining. *Wiley Interdiscip. Rev.-Data Mining Knowl. Disc.* **8**, 17 (2018)
14. Borkar, S., Rajeswari, K.: Predicting students academic performance using education data mining. *Int. J. Comput. Sci. Mobile Comput.* **2**(7), 273–279 (2013)
15. Buniyamin, N., bin Mat, U., Arshad, P.M.: Educational data mining for prediction and classification of engineering students achievement. In: 2015 IEEE 7th International Conference on Engineering Education (ICEED), pp. 49–53 (2015)
16. Cabestrero, R., Quiros, P., Santos, O.C., Salmeron-Majadas, S., Urias-Rivas, R.: Some insights into the impact of affective information when delivering feedback to students. *Behav. Inf. Technol.* **37**, 1–12 (2018)
17. Cengiz, M., Birant, K.U., Yildirim, P., Birant, D.: Development of an interactive game-based learning environment to teach data mining. *Int. J. Eng. Educ.* **33**, 1598–1617 (2017)
18. Charitopoulos, A., Rangoussi, M., Koulouriotis, D.: Educational data mining and data analysis for optimal learning content management: applied in moodle for undergraduate engineering studies. In: 2017 IEEE Global Engineering Education Conference (EDUCON), pp. 990–998 (2017)
19. Chau, V.T.N., Loc, P.H., Tran, V.T.N.: A robust mean shift-based approach to effectively clustering incomplete educational data. In: International Conference on Advanced Computing & Applications (2016)
20. Chen, H., Dai, Y., Feng, Y., Jiang, B., Xiao, J., You, B.: Construction of affective education in mobile learning: the study based on learner's interest and emotion recognition. *Comput. Sci. Inf. Syst.* **14**(3), 685–702 (2017)
21. Chen, J., Zhao, J.: An educational data mining model for supervision of network learning process. *Int. J. Emerg. Technol. Learn.* **13**, 67–77 (2018)
22. Conijn, R., Van den Beemt, A., Cuijpers, P.: Predicting student performance in a blended MOOC. *J. Comput. Assist. Learn.* **34**, 615–628 (2018)

23. Daghestani, L.F., Ibrahim, L.F., Al-Towirgi, R.S., Salman, H.A.: Adapting gamified learning systems using educational data mining techniques. *Comput. Appl. Eng. Educ.* **28**(3), 568–589 (2020)
24. Ducange, P., Pecori, R., Sarti, L., Vecchio, M.: Educational big data mining: how to enhance virtual learning environments, pp. 681–690 (2016)
25. Fernandez, D.B., Lujan-Mora, S.: Comparison of applications for educational data mining in engineering education. In: 2017 IEEE World Engineering Education Conference (EDUNINE) (2017)
26. FerreiraCmello, R., André, M., Pinheiro, A., Costa, E., Romero, C.: Text mining in education. *Wiley Interdiscip. Rev. Data Mining Knowl. Disc.* **9**(6), e1332 (2019)
27. Guo, B., Zhang, R., Xu, G., Shi, C., Yang, L.: Predicting students performance in educational data mining, pp. 125–128 (2015)
28. Gursoy, M.E., Inan, A., Nergiz, M.E., Saygin, Y.: Privacy-preserving learning analytics: challenges and techniques. *IEEE Trans. Learn. Technol.* **10**(1), 68–81 (2016)
29. Guruler, H., Istanbullu, A., Karahasan, M.: A new student performance analysing system using knowledge discovery in higher educational databases. *Comput. Educ.* **55**, 247–254 (2010)
30. Han, W., Jun, D., Xiaopeng, G., Kangxu, L.: Supporting quality teaching using educational data mining based on openedx platform, pp. 1–7 (2017)
31. Hao, Y.: Research on the formation rules and educational countermeasures of college students' socialist core values under big data. *Chimica OGGI-Chem. Today* **36**(6), 641–643 (2018)
32. Hegde, V., Prageeth, P.P.: Higher education student dropout prediction and analysis through educational data mining. In: 2018 2nd International Conference on Inventive Systems and Control (ICISC) (2018)
33. Hernandezblanco, A., Herreraflores, B., Tomas, D., Navarrocolorado, B.: A systematic review of deep learning approaches to educational data mining. *Complexity* 1–22 (2019)
34. Huang, C.T., Lin, W.T., Wang, S.T., Wang, W.S.: Planning of educational training courses by data mining: using China motor corporation as an example. *Exp. Syst. Appl.* **36**(3), 7199–7209 (2009)
35. Iam-On, N., Boongoen, T.: Generating descriptive model for student dropout: a review of clustering approach. *Hum.-Centric Comput. Inf. Sci.* **7**(1), 1–24 (2017)
36. Injatad, M., Moubayed, A., Nassif, A.B., Shami, A.: Systematic ensemble model selection approach for educational data mining. *Knowl.-Based Syst.* **200**, 47–62 (2020)
37. Jaber, M., Wood, P.T., Papapetrou, P., Gonzalezmarcos, A.: A multi-granularity pattern-based sequence classification framework for educational data, pp. 370–378 (2016)
38. Jacob, J., Jha, K., Kotak, P., Puthran, S.: Educational data mining techniques and their applications. In: International Conference on Green Computing & Internet of Things (2016)
39. Jalota, C., Agrawal, R.: Analysis of educational data mining using classification. In: 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), pp. 243–247 (2019)
40. Jena, R.K.: Sentiment mining in a collaborative learning environment: capitalising on big data. *Behav. Inf. Technol.* **38**(9), 986–1001 (2019)
41. Jones, K.M., Rubel, A., LeClere, E.: A matter of trust: higher education institutions as information fiduciaries in an age of educational data mining and learning analytics. *J. Assoc. Inf. Sci. Technol.* **71**(10), 1227–1241 (2020)

42. Kapelner, A., Soterwood, J., Nessaiver, S., Adlof, S.: Predicting contextual informativeness for vocabulary learning. *IEEE Trans. Learn. Technol.* **11**, 13–26 (2018)
43. Karthikeyan, V.G., Thangaraj, P., Karthik, S.: Towards developing hybrid educational data mining model (HEDM) for efficient and accurate student performance evaluation. *Soft Comput.* **24**(24), 18477–18487 (2020)
44. Kaur, P., Singh, M., Josan, G.S.: Classification and prediction based data mining algorithms to predict slow learners in education sector. *Proc. Comput. Sci.* **57**, 500–508 (2015)
45. Kausar, S., Huahu, X., Hussain, I., Wenhao, Z., Zahid, M.: Integration of data mining clustering approach in the personalized E-learning system. *IEEE Access* **6**, 1 (2018)
46. Khasanah, A.U., Harwati: A comparative study to predict student's performance using educational data mining techniques (2017)
47. Kim, B.H., Vizitei, E., Ganapathi, V.: GritNet: student performance prediction with deep learning (2018)
48. Kovalev, S., Kolodenkova, A., Muntyan, E.: Educational data mining: current problems and solutions. In: 2020 V International Conference on Information Technologies in Engineering Education (Inforino) (2020)
49. Li, C., Zhou, H.: Enhancing the efficiency of massive online learning by integrating intelligent analysis into MOOCs with an application to education of sustainability. *Sustainability* **10**(2), 16 (2018)
50. Li, Y., Gou, J., Fan, Z.: Educational data mining for students' performance based on fuzzy C-means clustering. *J. Eng.* **2019**(11), 8245–8250 (2019)
51. Lim, S., Tucker, C.S., Jablakow, K., Pursel, B.: A semantic network model for measuring engagement and performance in online learning platforms. *Comput. Appl. Eng. Educ.* **26**(5), 1481–1492 (2018)
52. Liu, Q., Zhang, S., Wang, Q., Chen, W.: Mining online discussion data for understanding teachers reflective thinking. *IEEE Trans. Learn. Technol.* **11**(2), 243–254 (2017)
53. Luna, J.M., Pechenizkiy, M., Del Jesus, M.J., Ventura, S.: Mining context-aware association rules using grammar-based genetic programming. *IEEE Trans. Cybern.* **48**(11), 3030–3044 (2018)
54. Luo, Y., Zhou, G., Li, J., Xiao, X.: A MOOC video viewing behavior analysis algorithm. *Math. Probl. Eng.* **2018**, 7 (2018)
55. Maclellan, C.J., Harpstead, E., Patel, R., Koedinger, K.R.: The apprentice learner architecture: Closing the loop between learning theory and educational data. In: 9th International Conference on Educational Data Mining - EDM 2016 (2016)
56. Manhaes, L.M.B., da Cruz, S.M.S., Zimbrao, G.: The impact of high dropout rates in a large public brazilian university a quantitative approach using educational data mining. In: CSEDU 2014 - Proceedings of the 6th International Conference on Computer Supported Education, pp. 124–129, January 2014
57. Menon, A., Gagliani, S., Haynes, M.R., Tackett, S.: Using “big data” to guide implementation of a web and mobile adaptive learning platform for medical students. *Med. Teach.* **39**, 975–980 (2017)
58. Merceron, A.: Educational data mining/learning analytics: methods, tasks and current trends. pp. 101–109 (2015)
59. Merceron, A., Yacef, K.: Educational data mining: a case study, pp. 467–474 (2005)
60. Injadat, M., Moubayed, A., Nassif, A.B., Shami, A.: Systematic ensemble model selection approach for educational data mining. *Knowl.-Based Syst.* **200**, 105992 (2020)

61. Moloo, R.K., Khedo, K.K., Prabhakar, T.V.: Critical evaluation of existing audio learning systems using a proposed TOL model. *Comput. Educ.* **117**, 102–115 (2018)
62. Moscoso, O., Vizcaíno, M., Luján-Mora, S.: Evaluation of methods and algorithms of educational data mining. In: 2017 Research in Engineering Education Symposium (2017)
63. Moscoso-Zea, O., Andres-Sampedro, Lujan-Mora, S.: Datawarehouse design for educational data mining. In: International Conference on Information Technology Based Higher Education & Training (2016)
64. Mou, C., Zhou, Q., Zou, X.: Understanding and predicting poor performance of computer science students from short time series test results. *Int. J. Eng. Educ.* **33**(6), 1803–1814 (2017)
65. Mushtaq, H., Siddique, I., Malik, B.H., Ahmed, M., Butt, U.M., Ghafoor, R.M.T., Zubair, H., Farooq, U.: Educational data classification framework for community pedagogical content management using data mining. *Int. J. Adv. Comput. Sci. Appl.* **10**(1), 329–338 (2019)
66. Urbina Najera, A.B., De La Calleja, J., Medina, M.A.: Associating students and teachers for tutoring in higher education using clustering and data mining. *Comput. Appl. Eng. Educ.* **25**(5), 823–832 (2017)
67. do Nascimento, R.L.S., das Neves Junior, R.B., de Almeida Neto, M.A., de Araujo Fagundes, R.A.: An Application of Regressors in Predicting School Dropout, Educational Data Mining (2018)
68. Onan, A.: Mining opinions from instructor evaluation reviews: a deep learning approach. *Comput. Appl. Eng. Educ.* **28**(1), 117–138 (2020)
69. Osei-Bryson, K.M.: Towards supporting expert evaluation of clustering results using a data mining process model. *Inf. Sci.* **180**(3), 414–431 (2010)
70. Antonenko, P.D., Toy, S., Niederhauser, D.S.: Using cluster analysis for data mining in educational technology research. *Educ. Technol. Res. Dev.* **60**(3), 383–398 (2012)
71. Penteado, B.E., Paiva, P.M.P., Morettin-Zupelari, M., Isotani, S., Ferrari, D.V.: Toward better outcomes in audiology distance education: an educational data mining approach. *Am. J. Audiol.* **27**(3S), 513–525 (2018)
72. Predic, B., Dimic, G., Rancic, D., Strbac, P., Macek, N., Spalevic, P.: Improving final grade prediction accuracy in blended learning environment using voting ensembles. *Comput. Appl. Eng. Educ.* **26**, 2294–2306 (2018)
73. Pristyanto, Y., Pratama, I., Nugraha, A.F.: Data level approach for imbalanced class handling on educational data mining multiclass classification. In: 2018 International Conference on Information and Communications Technology (ICOIACT) (2018)
74. Romero, C., Ventura, S.: Educational data mining and learning analytics: an updated survey. *Wiley Interdiscip. Rev.: Data Mining Knowl. Disc.* **12** e1355 (2019)
75. Romero, C., Ventura, S.: Educational data science in massive open online courses. *Wiley Interdiscip. Rev.: Data Mining Knowl. Disc.* **7**(1), e1187 (2017)
76. Rosado, J.T., Payne, A.P., Rebong, C.B.: eMineProve: educational data mining for predicting performance improvement using classification Method. In: Iop Conference, pp. 012018 (2019)
77. Kelly, J.D.O., Menezes, A.G., de Carvalho, A.B., Montesco, C.A.: Supervised learning in the context of educational data mining to avoid university students dropout, pp. 207–208 (2019)
78. Shrestha, R.M., Orgun, M.A., Busch, P.: Offer acceptance prediction of academic placement. *Neural Comput. Appl.* **27**(8), 2351–2368 (2016)

79. Srivastava, S., Karigar, S., Khanna, R., Agarwal, R.: Educational data mining: classifier comparison for the course selection process. In: 2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE), pp. 1–5 (2018)
80. Tanes, Z., Arnold, K.E., King, A.S., Remnet, M.A.: Using Signals for appropriate feedback: perceptions and practices. *Comput. Educ.* **57**(4), 2414–2422 (2011)
81. Tang, S., Peterson, J.C., Pardos, Z.A.: Deep neural networks and how they apply to sequential education data. In: 3rd Annual ACM Conference on Learning at Scale, L@S 2016, 25 April 2016–26 April 2016, pp. 321–324 (2016)
82. Tasnim, N., Paul, M.K., Sattar, A.S.: Identification of drop out students using educational data mining (2019)
83. Toivonen, T., Jormanainen, I.: Evolution of decision tree classifiers in open ended educational data mining (2019)
84. Tseng, C.W., Chou, J.J., Tsai, Y.C.: Text mining analysis of teaching evaluation questionnaires for the selection of outstanding teaching faculty members. *IEEE Access* **6**, 72870–72879 (2018)
85. Villegas-Ch, W., Lujan-Mora, S., Buenano-Fernandez, D.: Towards the integration of business intelligence tools applied to educational data mining. In: 2018 IEEE World Engineering Education Conference (EDUNINE) (2018)
86. Waheed, H., Hassan, S.U., Aljohani, N.R., Wasif, M.: A bibliometric perspective of learning analytics research landscape. *Behav. Inf. Technol.* **37**(10–11), 941–957 (2018)
87. Waheed, H., Hassan, S.U., Aljohani, N.R., Hardman, J., Alelyani, S., Nawaz, R.: Predicting academic performance of students from VLE big data using deep learning models. *Comput. Hum. Behav.* **104**, 106189 (2020)
88. Wang, F.H.: An exploration of online behaviour engagement and achievement in flipped classroom supported by learning management system. *Comput. Educ.* **114**, 79–91 (2017)
89. Wang, L., Hu, G., Zhou, T.: Semantic analysis of learners' emotional tendencies on online MOOC education. *Sustainability* **10**(6), 1921 (2018)
90. Wook, M., Yusof, Z.M., Nazri, M.Z.A.: Educational data mining acceptance among undergraduate students. *Educ. Inf. Technol.* **22**, 1195–1216 (2017)
91. Xiong, F., Zou, K., Liu, Z., Wang, H.: Predicting learning status in MOOCs using LSTM. In: 2019 ACM Turing Celebration Conference - China, ACM TURC 2019, 17 May 2019–19 May 19 2019
92. Yahya, A.A., Mohammed, F.A., Osman, A.: A novel use of educational data mining to inform effective management of academic programs. *Life-long Learn.* **100**(130), 130 (2019)
93. affar, M., Hashmani, M.A., Savita, K.S.: Performance analysis of feature selection algorithm for educational data mining. In: IEEE Conference on Big Data & Analytics (2017)
94. Zaffar, M., Hashmani, M.A., Savita, K.S.: Comparing the performance of FCBF, Chi-Square and relief-F filter feature selection algorithms in educational data mining, pp. 151–160, June 2019
95. Zhang, Y., Shah, R., Chi, M.: Deep learning + student modeling + clustering: a recipe for effective automatic short answer grading, pp. 562–567 (2016)
96. Zhou, Q., Quan, W., Zhong, Y., Xiao, W., Mou, C., Wang, Y.: Predicting high-risk students using Internet access. *Knowl. Inf. Syst.* **55**(2), 393–413 (2018)



Image Classification with A-MnasNet and R-MnasNet on NXP Bluebox 2.0

Prasham Shah and Mohamed El-Sharkawy^(✉)

IoT Collaboratory at IUPUI, Department of Electrical and Computer Engineering,
Purdue School of Engineering and Technology, Indianapolis, IN 46202, USA
{pashah,melshark}@purdue.edu

Abstract. Bluebox 2.0 by NXP Semiconductors, which has goal of enabling autonomy in vehicles for ADAS applications, is used to enhance car capabilities to perform sensor fusion and run AI algorithms. It focuses on sensor data coming from radars, lidars, and cameras. This research focuses on enabling computer vision application, Image Classification, by implementation of Convolutional Neural Networks in Bluebox 2.0. In this paper, two CNN architectures namely A-MnasNet and R-MnasNet have implemented on Bluebox 2.0. These models have been derived by Design Space Exploration of MnasNet, a CNN architecture, proposed by Google Brain team in 2019. These models have been trained and tested on CIFAR-10 dataset. The model size and accuracy of A-MnasNet are 11.6 MB and 96.89% and that of R-MnasNet are 3 MB and 91.13% respectively. They outperform the MnasNet architecture which has an accuracy of 80.8% and a model size of 12.7 MB. These neural networks can also be used to perform other computer vision applications.

Keywords: Convolutional Neural Networks (CNNs) · Computer vision · Image classification · A-MnasNet · R-MnasNet · NXP Bluebox 2.0 · MnasNet

1 Introduction

CNNs enable computer vision applications like image classification, object detection, etc. The complexity of deployment of CNNs on the resource constrained hardware for applications that require extensive computational efficiency in terms of accuracy and latency plays a critical role in development of such models. As power, memory, size and other resources available on such embedded platforms are limited, designing mobile CNNs with state-of-the-art efficiency has become a prime focus of the research. Reducing the depth of CNN architectures and using computationally augmented convolutions have made the models more efficient in terms of computational overload.

A-MnasNet [1] and R-MnasNet [2] are new CNN architectures which are designed to work efficiently on resource constrained platforms. These architectures have been derived from MnasNet. New algorithms were implemented, by Design

Space Exploration of MnasNet, for better efficiency with a fair trade-off between model size and accuracy. These architectures were deployed on Bluebox 2.0 for Image Classification. RTMaps Studio was used for real-time implementation of these CNN architectures on NXP Bluebox 2.0.

This paper will explain the features of the A-MnasNet [1] and R-MnasNet [2] CNN architectures. It will explain how these new CNN architectures were proposed by Design Space Exploration of MnasNet. Furthermore, it will give an insight on how these new CNN models were used for Image Classification on Bluebox 2.0 using RTMaps Studio.

The next section, literature review, will give an insight on Bluebox 2.0 and RTMaps. It will discuss the hardware and software utilized for implementation of A-MnasNet [1] and R-MnasNet [2] on NXP Bluebox 2.0. Third section explains the features and significance of the implemented CNN architectures i.e. A-MnasNet and R-MnasNet. Section 4 will describe the hardware and software used for this research. Section 5 gives an insight on implementation setup and methodology. The next section will discuss the results obtained after this research followed by conclusion and future scope.

2 Literature Review

2.1 NXP Bluebox 2.0

BlueBox 2.0 is a hardware from NXP which is used for ADAS domain for sensor fusion and computer vision applications to enable autonomy in vehicles. Fig. 1 shows the NXP Bluebox 2.0 development hardware which is used for this research



Fig. 1. NXP BlueBox 2.0

High Level View of BlueBox Resources

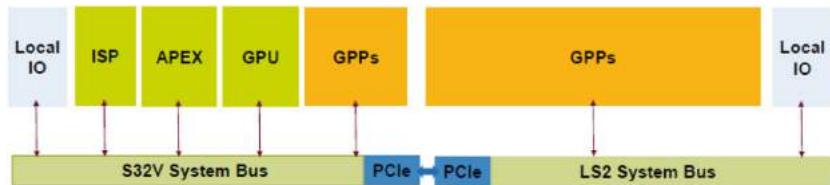


Fig. 2. High level view of NXP Bluebox 2.0 resources

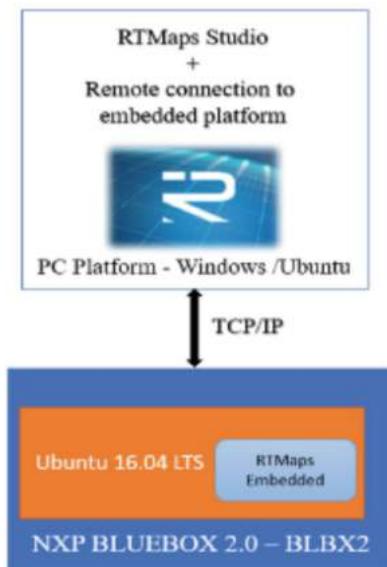
and Fig. 2 depicts its resources. It has three SoCs S32V234: vision processor, LS2084A: compute processor, and S32R274: radar microcontroller.

Figure 3 shows the connection between Bluebox 2.0 and RTMaps Studio. RTMaps Studio is used to deploy CNNs on Bluebox 2.0. It establishes a TCP/IP connection for deployment. It enables the deployment of CNNs through a python module. It gives the user flexibility of deploying CNNs with any deep learning framework like Tensorflow, PyTorch, etc.

Table 1. A-MnasNet architecture

A-MnasNet architecture						
Layers	Convolutions	t	c	n	s	
$32^2 \times 3$	Conv2d 3×3	-	32	1	1	
$112^2 \times 32$	SepConv 3×3	1	16	1	2	
$112^2 \times 16$	MBCConv3 3×3	3	24	3	2	
$112^2 \times 24$	Harmonious Bottleneck	2	36	1	1	
$56^2 \times 36$	MBCConv3 5×5	3	40	3	2	
$112^2 \times 40$	Harmonious Bottleneck	2	72	1	2	
$28^2 \times 72$	MBCConv6 5×5	6	80	3	2	
$112^2 \times 80$	Harmonious Bottleneck	2	96	4	2	
$14^2 \times 96$	MBCConv6 3×3	6	96	2	1	
$14^2 \times 96$	MBCConv6 5×5	6	192	4	1	
$7^2 \times 192$	MBCConv6 3×3	6	320	1	1	
$7^2 \times 320$	FC, Pooling			10		

t: expansion factor, c: number of output channels, n: number of blocks and s: stride

**Fig. 3.** RTMaps vonnection with Bluebox 2.0**Table 2.** R-MnasNet architecture

R-MnasNet architecture						
Layers	Convolutions	t	c	n	s	
$32^2 \times 3$	Conv2d 3×3	-	32	1	1	
$112^2 \times 32$	SepConv 3×3	1	16	1	2	
$112^2 \times 16$	MBCConv3 3×3	3	24	3	2	
$112^2 \times 24$	Harmonious Bottleneck	2	36	1	1	
$56^2 \times 36$	MBCConv3 5×5	3	40	3	2	
$112^2 \times 40$	Harmonious Bottleneck	2	72	1	2	
$28^2 \times 72$	MBCConv6 5×5	6	80	3	2	
$112^2 \times 80$	Harmonious Bottleneck	2	96	4	2	
$14^2 \times 96$	MBCConv6 3×3	6	96	2	1	
$112^2 \times 80$	Harmonious Bottleneck	2	192	1	2	
$112^2 \times 80$	Harmonious Bottleneck	2	96	4	2	
$14^2 \times 96$	MBCConv6 5×5	6	192	4	1	
$112^2 \times 80$	Harmonious Bottleneck	2	288	1	1	
$7^2 \times 192$	MBCConv6 3×3	6	320	1	1	
$7^2 \times 320$	FC, Pooling				10	

t: expansion factor, c: number of output channels, n: number of blocks and s: stride

3 Features of A-MnasNet and R-MnasNet

Table 1 and Table 2 show the architecture of A-MnasNet and R-MnasNet, respectively.

3.1 Convolutions

Convolutional layers are the most important part of CNNs. They are the backbone of CNNs. They are used to extract and learn features from the input image. The performance of CNNs depend on the type of convolutional layers used in the architecture. The accuracy and model size are greatly affected by these layers.

Different types of convolutions are used to extract features from an image or a video input. Depthwise Separable layers were used in MnasNet [3]. In order to extract features more efficiently, Harmonious Bottleneck Layers [4] were added to the architecture.

These convolutional layers extract features from the spatial dimensions along with the channel dimensions but it changes the scale along these dimensions as well. There is contraction-expansion of spatial dimensions while keeping the channel dimensions constant and expansion-contraction of channel dimensions while keeping the spatial dimensions constant. The computational cost of Harmonious Bottleneck Layers [4] is less than the depthwise separable convolutional layers. This strikes a decrease in the model size of the architecture and increases its accuracy. Comparison of Depthwise Separable Convolution Layer and Harmonious Bottleneck Layer is demonstrated in Fig. 4.

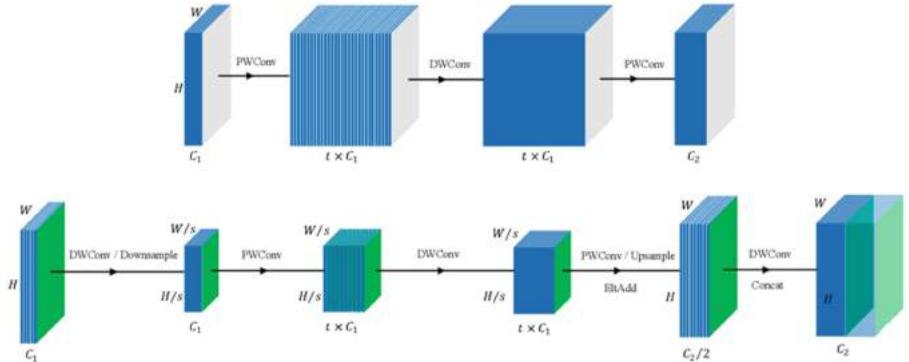


Fig. 4. Comparison of depthwise separable convolution layer and harmonious bottleneck layer [2].

The spatial size of input/output feature maps is $(H \times W)$, $C1/C2$ are input/output feature channels, $(K \times K)$ is the kernel size and s denotes stride.

The total cost of depthwise separable convolution is

$$(H \times W \times C1 \times K \times K) + (H \times W \times C1 \times C2) \quad (1)$$

The total cost of harmonious bottleneck layer [2] is

$$B/s^2 + (H/s \times W/s \times C1 + H \times W \times C2) \times K^2 \quad (2)$$

where, B is the computational cost of the blocks inserted between the spatial contraction and expansion operations. It is evident that by squeezing the channel expansion-contraction component and using a pair of spatial transformations yields a slimmed spatial size of wide feature maps in each stage which reduces the computational cost.

These layers were implemented in A-MnasNet [1] and R-MnasNet [2]. After implementing these layers, it was evident that the accuracy of the models increased and the size of the models decreased.

3.2 Activation Functions

For non-linearity, ReLU activation function was used in the MnasNet [3]. The problem with ReLU was its inability to preserve negative values. In order to rectify this, Mish [7] activation function is used in R-MnasNet. It is a different kind of softplus function. Mathematically, it can be represented as:

$$f(x) = x \cdot \tanh(\epsilon(x)) \quad (3)$$

where,

$$\epsilon(x) = \ln(1 + x) \quad (4)$$

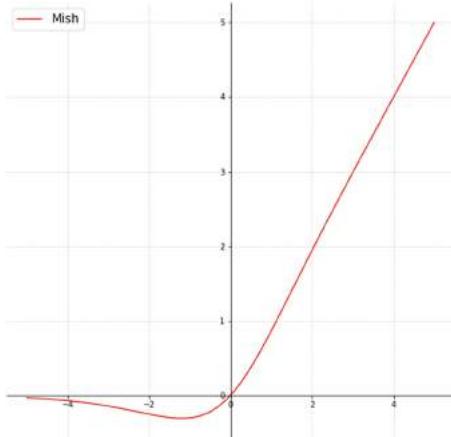


Fig. 5. Mish activation function

Mish [7] is shown in Fig. 5. It avoids saturation due to near zero gradients and has strong regularization effects. It outperforms all existing activation functions. It preserves small negative gradients. Implementation of this activation function results in effective optimization and generalization.

3.3 Data Augmentation

AutoAugment [5] was used to pre-process the data. It learns the best augmentation policies by using Reinforcement Learning. Each policy has five sub-policies and each sub-policy applies 2 image operations in sequence based on the probability of applying it and the magnitude of operation.

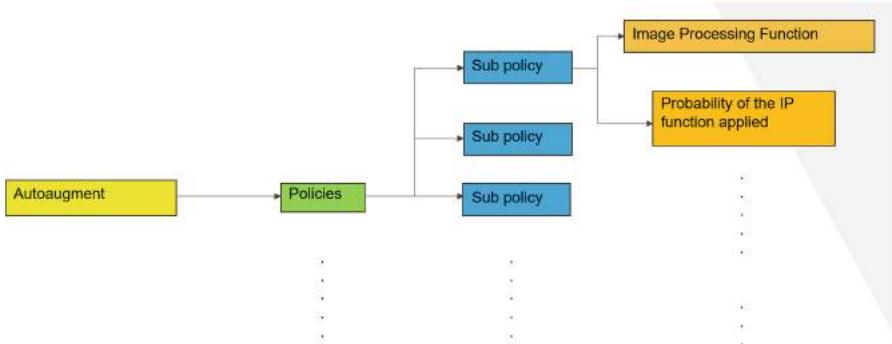


Fig. 6. AutoAugment

Figure 6 represents the augmentation process. The best policies are selected automatically by using reinforcement learning. The accuracy of the models are enhanced by pre-processing the data using AutoAugment [5].

3.4 Learning Rate Annealing or Scheduling

Learning rates determine rate of back-propagation during optimization of CNNs. Various learning rates are used to enhance the training process. Learning rates are reduced during training. Figure 7 depicts Step decay to be the best choice for scheduling. Hence, it was used for [1] and [2] CNN architectures. Variable learning rates of 0.1, 0.01 and 0.001 were used while training with CIFAR-10 [6] dataset.

3.5 Optimizers

Optimization is an important aspect of the CNNs. This process calculates the error using a loss function and then tries to minimize that error. There are different optimizers which are used for the optimization of CNN architectures.

RMSprop was used to train MnasNet [3]. SGD [8] was used to train R-MnasNet with momentum equal to 0.9. Learning rate scheduler was used while training the network.

4 Implementation on NXP BLUEBOX 2.0

This section will discuss the implementation of the proposed CNN architectures.

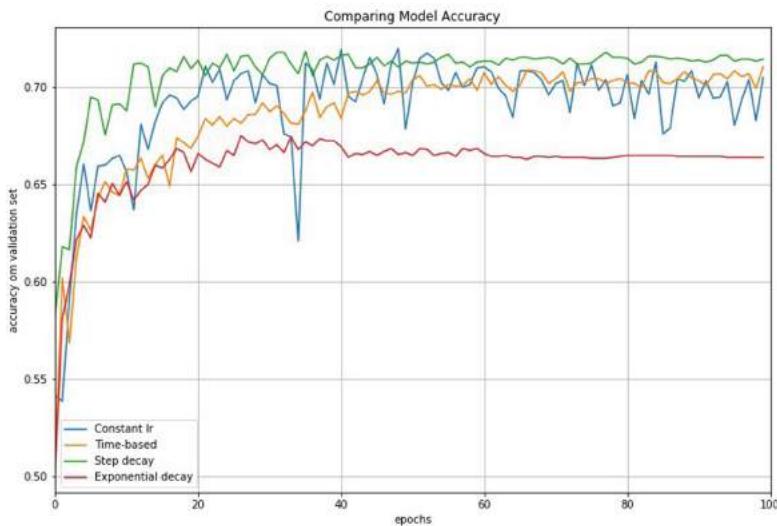


Fig. 7. Comparison of LR scheduling methods

4.1 Implementation Setup

Implementation was done by using RTMaps Studio. It provides an interface between NXP Bluebox 2.0 and the CNN architectures via a TCP/IP connection. The architecture is deployed using a python module. The process is illustrated in Fig. 8.

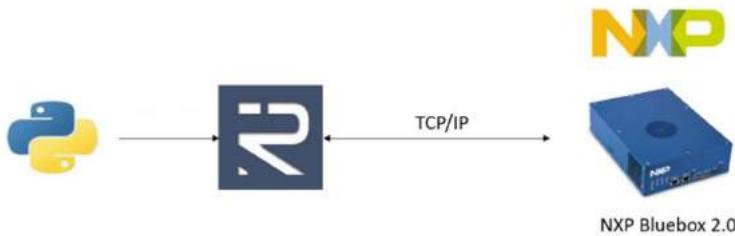


Fig. 8. Implementation on NXP Bluebox 2.0

4.2 Implementation Methodology

The CNN models were trained on CIFAR-10 dataset using PyTorch framework. After the training, the models were saved in a checkpoint file. These files were used to deploy A-MnasNet and R-MnasNet on NXP Bluebox 2.0 for Image Classification. Figure 9 demonstrates the implementation process.

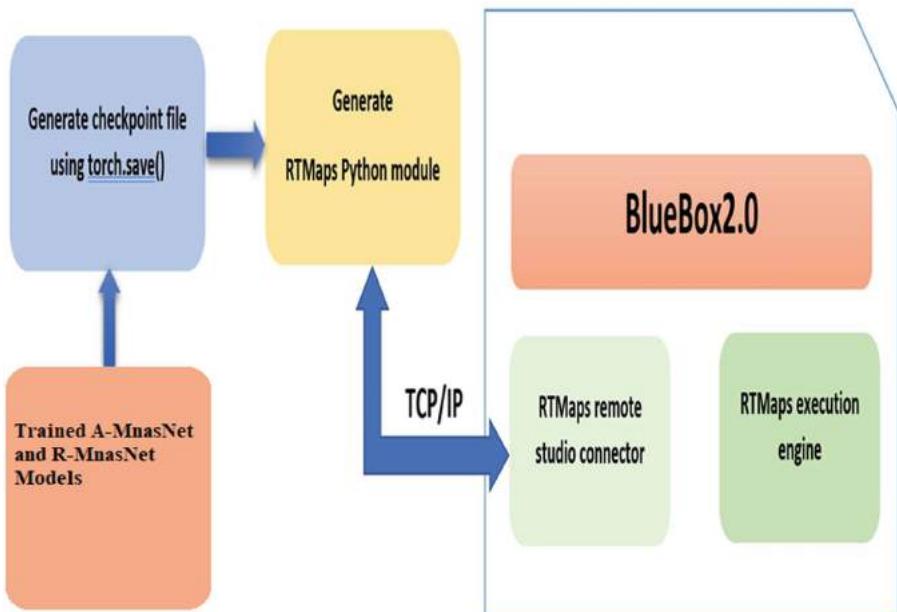


Fig. 9. Implementation methodology

The trained models were imported in the RTMaps Studio using its python component. The python module is shown in Fig. 10.



Fig. 10. Python component of RTMaps

The python component in RTMaps has a text editor that allows users to modify their code. It works due to the combination of three functions. They are Birth(), Core() and death().

- Birth(): used to initialize and define the parameters.
- Core(): used to import the CNN architectures
- Death(): used to stop the implementation.

Data transmission between RTMaps Studio and NXP Bluebox 2.0 is done by establishing a TCP/IP connection. The connection is established by assigning an IP address to the RTMaps Studio connector and the LS2 Ethernet port of NXP Bluebox 2.0. The trained models are then successfully deployed and used for Image Classification.

5 Results

A-MnasNet and R-MnasNet are used for Image Classification on NXP Bluebox 2.0. These models have an accuracy of 96.89% and 91.13% with a model size of 11.6 MB and 3 MB respectively. They outperform the baseline MnasNet architecture in terms of model size and accuracy. A comparison of these models is shown in Table 3.

Table 3. Comparison of models

Comparison of models		
Architecture	Model accuracy	Model size (in MB)
MnasNet	80.8%	12.7
A-MnasNet	96.89%	11.6
R-MnasNet	91.13%	3

These models were trained with CIFAR-10 [6] dataset on Aorus Geforce RTX 2080Ti GPU using PyTorch framework for 200 epochs. The data was divided into batch size of 128 for training set and batch size of 64 for validation set. Cross entropy loss function was used for the error calculations. Optimization was done using SGD optimizer with varying learning rates of 0.1, 0.01 and 0.001.

After training, the models were imported in the RTMaps Studio using its python component. The python module is shown in Fig. 10. A TCP/IP connection is used for data transmission between RTMaps Studio and NXP Bluebox 2.0. The models were successfully deployed on NXP Bluebox 2.0. They were tested for image classification. They were given input from the dataset. They were able to predict the object in the input images accurately. The console output of A-MnasNet and R-MnasNet are shown in Fig. 11 and Fig. 12, respectively.

Console X

```

local
No documentation found for component "inputl"
Start recording.
Info: Starting main thread 0x1398 for component python_v2_1
Info: component python_v2_1: Python Birth
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: plane
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: deer
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: car
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1:
Info: component python_v2_1: ship
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: plane
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: car

```


Fig. 11. Image classification using A-MnasNet

Console X

```

local
No documentation found for component "inputl"
Start recording.
Info: Starting main thread 0x1398 for component python_v2_1
Info: component python_v2_1: Python Birth
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: deer
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: cat
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1:
Info: component python_v2_1: car
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: plane
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: cat
Info: component python_v2_1: Predicted:
Info: component python_v2_1:
Info: component python_v2_1: ship

```


Fig. 12. Image classification using R-MnasNet

6 Conclusion

This paper demonstrates Image Classification on NXP Bluebox 2.0 using Convolutional Neural Networks. A-MnasNet [1] and R-MnasNet [2] which have been derived from MnasNet have been used for this Computer Vision application. These models, when trained on CIFAR-10 [6] dataset using Pytorch framework, have a validation accuracy of 96.89% and 91.13% with a model size of 11.6 MB and 3 MB respectively. They outperform the baseline MnasNet [3] architecture in terms of model size and accuracy. These models are enhanced with Harmonious Bottleneck Layers [4], Mish Activation Functions [7] and AutoAugment [5]. RTMaps Studio was used to deploy these architectures to NXP Bluebox 2.0 by establishing a TCP/IP connection. These models can also be used for other computer vision applications like Object Localization, Object Detection, Semantic Segmentation, etc. on NXP Bluebox 2.0 as well as other mobile or embedded platforms.

7 Future Scope

These architectures were trained and tested on CIFAR-10 [6] from scratch. Transfer Learning can be used to improve the efficiency of these models. This architectures can be also be used for other computer vision applications like object detection, object recognition, semantic segmentation, etc. Deep Compression is a technique, which is used to reduce the model parameters. It uses pruning, quantization and Huffman encoding on the network to compress the network. This technique has been used on state-of-the-art architectures to reduce their size and has successfully accomplished that. This technique could be used to further enhance A-MnasNet [1] and R-MnasNet [2].

References

- Shah, P., El-Sharkawy, M.: A-MnasNet: augmented MnasNet for computer vision. In: IEEE 63rd International Midwest Symposium on Circuits & Systems (MWSCAS 2020), Springfield, Massachusetts, 9–12 August 2020
- Shah, P., El-Sharkawy, M.: R-MnasNet: reduced MnasNet for computer vision. In: International IOT, Electronics and Mechatronics Conference (IEMTRONICS 2020), Vancouver, Canada, 9–12 September 2020
- Tan, M., Chen, B., et al.: MnasNet: Platform-Aware Neural Architecture Search for Mobile, [arXiv:1807.11162v3](https://arxiv.org/abs/1807.11162v3) [cs.CV], 29 May 2019. Accessed 14 Apr 2021
- Li, D., Zhou, A., Yao, A.: HBONet: Harmonious Bottleneck on Two Orthogonal Dimensions, [arXiv:1908.03888v1](https://arxiv.org/abs/1908.03888v1) [cs.CV], 11 August 2019. Accessed 14 Apr 2021
- Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: AutoAugment: Learning Augmentation Strategies from Data, [arXiv:1805.09501v3](https://arxiv.org/abs/1805.09501v3) (2019). Accessed 14 Apr 2021
- <https://www.cs.toronto.edu/~kriz/cifar.html>. Accessed 14 Apr 2021
- Misra, D.: Mish: A Self Regularized Non-Monotonic Neural Activation Function, arXiv preprint [arxiv:1908.08681](https://arxiv.org/abs/1908.08681) (2019). Accessed 14 Apr 2021
- Ruder, S.: An overview of gradient descent optimization algorithms, arXiv Preprint [arXiv:1609.04747](https://arxiv.org/abs/1609.04747) (2016). Accessed 14 Apr 2021



CreditX: A Decentralized and Secure Credit Platform for Higher Educational Institutes Based on Blockchain Technology

Romesh Liyanage^(✉), D. P. P. Jayasinghe, K. T. Uvindu Sanjana, H. B. D. R. Pearson, Disni Sriyaratna, and Kavinga Abeywardena

SLIIT University, Malabe 10115, Sri Lanka
{disni.s,kavinga.y}@sliit.lk

Abstract. The current flow of education and industry needs, consistent convergence among students, Higher Education Institutes (HEIs) and industrial organizations. Though, maintaining the credibility and verifiability of students' credit information is one of the vital priorities for HEIs, the current credit systems used by most of the HEIs are not sound, and lack effective means of verification. Most of these HEIs, currently utilize systems of centralized and monolithic nature which possesses limitations in availability, security, and scalability domains. Therefore, there exists an absolute requirement for a more secure, efficient, and scalable means of student's credit information management. Blockchain technology empowers the formation of a decentralized domain where data and transactions are not heavily influenced by any outsider association. Every transaction is logged in a public ledger in a verifiable and perpetual manner allowing the visibility to all the parties involved. This enables the creation of a system that is not feasible to penetrate. Therefore, we propose a more credible and innovative solution called CreditX, a decentralized and a secure credit platform for HEIs to collaborate, share and transfer credit data among HEIs, students and industrial organizations while establishing trust.

Keywords: Blockchain · Decentralization · Higher education institutes

1 Introduction

1.1 Background

Most of the organizations at the moment, use centralized software systems [3] to maintain their data. These can be employee data, transaction data, student credit information, etc. Even today, centralized, and monolithic software systems are the industry standard and the go-to solution for any organization looking to digitalize their data management process. This is due to the short-term advantages of centralization such as simplified IT infrastructure [4]. Higher education institutes are no exception to this practice. Majority of higher education institutions including Sri Lanka Institute of Information Technology (SLIIT), utilizes information systems of centralized nature to maintain students' credit

and grades [1]. Though there are advantages to centralization, for platforms of this nature, the advantages of decentralization [3] outnumber them from a large margin. One of the main shortcomings of centralized systems being its' integral reliance of a central server, if for any reason the central server becomes unavailable, the entire system will be of no use for the duration. Another key shortcoming of centralization is security. Especially the entire system being highly dependent on a central server curves the path for a single point of failure. It opens opportunities for adversaries to penetrate the system. The information stored in these systems being open for modifications, the risk always exists of malicious intruders penetrating the system and modifying the information. There have been instances in recent history, where higher education institution personal have been accused of tampering with student grades for their vested interests [5]. Incidents of this nature not only tarnish the reputation of the institution in the academic world but also extirpate the confidence the students have in the institution. Maintaining the credibility and verifiability of students' credit information is one of the highest priorities for a higher education institute. Therefore, considering the limitations of the current centralized and monolithic based approach in security, integrity, availability and scalability domains, there exist an absolute requirement for a more secure, efficient and scalable means of managing students' credit information.

Considering all the above observations, we concluded that a blockchain [2] based, decentralized credit platform would mitigate the limitations of centralization, and provide all the benefits inherently associated with blockchain and decentralization technologies. Therefore, by default the system will be highly robust, transparent, scalable, and secure [2]. Blockchains are designed as a decentralized database that functions as a distributed ledger [2]. These blockchain ledgers record and store data in blocks, which are organized in a chronological sequence and are linked through cryptographic proofs. Since the blockchain data is often stored in a distributed network of nodes, the system and data are highly resilient to technical failures and malicious attacks [2]. Each network node can replicate and store a copy of the database and due to this, there will not be a single point of failure. A credit platform of this nature would provide reassurance to all the stakeholders of the institution, that the data and the transactions are secure. Furthermore, blockchains' distributed architecture and the transaction ledger would ensure that the transactions and the data of the system are immutable.

Our proposed CreditX platform mainly contains four components. The first component focuses on the backend logic of the proposed platform, which involves development and optimization of the smart contracts which are to be deployed in the blockchain network. The second component focuses on the blockchain client API implementation and the handling of data transfers between the blockchain network and the application. The third component focuses on how the higher security inherently associated with blockchain technology can be further enhanced and how to provide proper user authentication for the system. The final component of the proposed platform focuses on the implementation of a private deployment of a blockchain network. It investigates automated rule generation for optimal node configurations.

1.2 Literature Survey

As discussed above in the introduction, current systems harbour many limitations due to their nature of centralization. Mention below are some of the related work in different domains. They include both centralized systems and decentralized systems which are based on blockchain technology.

“Centralized Educational Institution Management System” [6] is a report by three researchers of BRAC University, proposing a centralized educational management solution for the educational sector of Bangladesh. These researchers propose that the central database will facilitate all the data transfers among the servers. Though this approach is practical when the data needs to be controlled centrally, this practice depicts a vast number of disadvantages such as this will cause higher security and privacy risks, prone to failures, longer access times for users who are far from the server and, single point of failure.

“School Management System for Government Schools in Ethiopia using Distributed a Database” [7], is an international journal of Engineering Trends and Technology (IJETT) published in 2018. In this journal they propose a system in order to overcome all the complex activities related to the school management. Though this solution fulfills the requirement, designing and maintaining a distributed database is prone to many challenges as proven by CAP theorem hypothesis [8].

The National University of La Plata (UNLP) has initiated developing a blockchain based platform for storing educational records and assessing academic achievements [9, 10] but no further details have been revealed. The same approach was adopted by the Argentinian College CESYT [11] and it too has not been revealed or a prototype has been published so far.

Sony Global Education [12] is an educational record storage platform based on blockchain technology which allows students to record their educational experiences. However, when analyzing the actual product according to the requirements [13], not only does the platform allow academic experiences to be recorded, but also informal non-academic experiences as well.

In a study published in 2018 a group of scholars have proposed a blockchain based global higher education credit platform named “EduCTX” [1]. This platform is based on the concept of the European Credit Transfer and Accumulation System (ECTS) [14]. It constitutes a globally trusted, decentralized higher education credit and grading system that can offer a globally unified viewpoint for students and higher education institutions, as well as for other potential stakeholders such as companies, institutions, and organizations. This proposed platform comes with all the advantages and disadvantages associated with blockchain applications. In the paper, the system is proposed only as a proof of concept. Therefore, neither an open-source or commercial working product is available. With this approach, all the higher education institutions have to be in a unified viewpoint. Consequently, the proposed system doesn’t allow an institution to maintain its grading and credit data independently. Furthermore, the paper doesn’t mention much detail on how or which technologies were used to provide user authentication and access control nor the frontend web application and the client API. However, when critically reviewing and analyzing the process which has to be followed by users as explained in

the paper, it can be safely deduced that the usability of the system is very low. According to a systematic review done by Ali Almmary on blockchain based applications in education [15] it is a must to implement the frontend web application that interacts with users to be more user friendly as the majority of the users lack the clarity and technical knowledge to comprehend the complexity behind blockchain technology.

Another group of scholars has proposed an Education-Industry cooperative system based on the Hyperledger framework [16]. Here their main object was to utilize decentralization and non-tampering features of the blockchain technology to eliminate the substantive information asymmetry which exists between higher education institutes and employing companies. When examining the security aspect of the proposed system, the paper does not provide much detail on the authentication model being utilized. The high-level architecture of the system illustrates that an authentication model is maintained on top of the Hyperledger. The paper states that validated users can interact with the system after being authenticated and obtaining a certificate from the Hyperledger Fabric CA server. However, one of the shortcomings of this proposed approach is that maintaining a master node (Node which contains “Hyperledger Fabric CA Server” and “Hyperledger Fabric Order”) and only one of that, from a security standpoint, could expose the system to a single point of failure.

When considering blockchain based platforms in other domains, Leslie Mertz has published a literature [17] proposing a system which utilizes blockchain technology to store and maintain patient data. In that the author points out the need for a distributed solution as transferring data for backup and redundancy purposes from one hospital or clinic to another can be challenging since patients usually get the service of multiple clinics and doctors depending on the needs. The paper also states that this application of blockchain technology in the healthcare industry could improve delicate processes such as insurance claim.

Blockchain solutions are also popular in the supply chain management industry due to the traceability aspect it provides. Denissolt and the team have published a paper [18] in 2019 about a blockchain implementation to analyze carbon footprint in food supply chain processes. The paper proposes a system to track and store details of the carbon footprint that occurs in food production and transportation.

1.3 Research Gap

When taking into consideration all the readings illustrated in the literature survey, it is clearly distinguishable that there is a large number of limitations present in the available literature. From one aspect, if we take into consideration the information systems of centralized nature currently being used, by default they inherit all the shortcomings associated with centralization. From the other aspect, when we investigate the attempts that have been made so far, to utilize blockchain technology in information systems, the literature survey presented above distinctly demonstrates that the work is still in a primitive stage. Most of those literatures only provide prototypes as proof of concept and so far as of this writing none has been released either as an open-source or a commercial product. Furthermore, even though the general architecture of the systems is illustrated, they don't provide much detail on the implementation of the blockchain network, client

API to transfer data between frontend and blockchain and how required security, user authentication and scalability was achieved.

Table 1 provided below illustrates what our proposed platform offers when compared to other solutions currently being used, with respect to the overall architecture, frontend application and client API implementation and the blockchain backend implementation.

Table 1. Research gap 1

Features	CEIMS	ESMS	EICS	EduCTX	CreditX
Relatively Complex Initial Implementation	✗	✗	✗	✓	✓
Data stored in system is immutable	✗	✗	✓	✓	✓
No single point of failure	✗	✗	✗	✓	✓
Not entirely dependent on a central server	✗	✗	✓	✓	✓
Transactions are transparent to all parties involved	✗	✗	✗	✓	✓
High usability and user-friendliness	✓	✓	✗	✗	✓
Data stored in system is immutable	✗	✗	✓	✓	✓
Works on a local private network	✗	✗	✗	✗	✓
High traceability of system operations	✗	✗	✓	✓	✓
Applied in working solution	✓	✓	✗	✗	✓
Not controlled by a single authority	✗	✗	✓	✓	✓
Not dependent on third party service providers	✗	✗	✗	✗	✓
Not feasible to tamper records	✗	✗	✓	✓	✓

Table 2 provided below illustrates what our proposed platform offers in terms of scalability when compared to other solutions currently being used.

Table 2. Research gap 2

Features	CEIMS	ESMS	EICS	EduCTX	CreditX
Vertical scaling	✓	✓	✓	✓	✓
Horizontal Scaling	✗	✗	✓	✓	✓
No bottlenecks when traffic spikes	✗	✗	✓	✓	✓
Define a set of rules for node configuration	✗	✗	✗	✓	✓
Provide GUIs to setup and manage nodes	✗	✗	✗	✗	✓
Indicates status of nodes with GUI support	✗	✗	✗	✗	✓
Local private network	✗	✗	✗	✗	✓

When looking into the security, and user authentication aspects of the currently used systems, most of them either utilize HTTP Basic authentication service, HTTP Digest authentication service or OAuth authentication service [19].

HTTP basic authentication is the standard method of access control provided by most major browsers. Basic authentication is supported at the HTTP level by most web servers and requires little or no development effort to implement. However, because basic authentication does not provide protection of the user id and the password during transmission from a user's computer to the web server, it leaves the opportunity for an adversary to intercept the information. Furthermore, basic authentication is somewhat less vulnerable under HTTPS. However, it primarily lacks behind other authentication models because of the necessity to submit the credentials with each request [20].

HTTP digest authentication was developed in order to mitigate the limitations of HTTP basic authentication. Digest authentication is an implementation of MD5 cryptographic hashing. The Nonce parameter is one of the most important attributes of this implementation. This parameter can prevent relay attacks. According to the digest authentication strategy, basic credentials value (username and password) is not sent to the server in native form but in a hashed form. So that during the transfer, credentials are protected against attacks [21]. However, digest authentication is not immune to all

ranges of attacks. There are several notable drawbacks. For instance, digest authentication is still vulnerable to man in the middle attacks. Where a malicious attacker could tell clients to use basic access authentication or legacy RFC2069 digest access authentication mode. To extend this further digest access authentication provides no mechanism for clients to verify the server's identity. Furthermore, digest authentication prevents the use of strong password hashes such as bcrypt [22] when storing passwords [23].

The most commonly used authentication service at the moment is OAuth 2.0 based authentication [19]. This framework enables applications to obtain limited access to user accounts on an HTTP service, such as Facebook, Google, GitHub, and DigitalOcean. It works by delegating user authentication to the service that hosts the user account and authorizing third-party applications to access the user accounts [24]. From a security standpoint, OAuth 2.0 was able to mitigate many of the limitations and vulnerabilities associated with other authentication services. At the moment OAuth 2.0 is the most secure data sharing standard on the market. The two-factor nature and use of tokenization prevent the single factor disclosure accounts [25]. Furthermore, OAuth 2.0 is immune to database flooding [19]. However, OAuth 2.0 also presents with another set of completely different issues. For an instance, if we access any website using Google or Facebook etc., and due to some issue, Facebook or Google blocks that particular website, we will lose access to that website. These emphasize the complications which arise when we are entirely dependent on third party authentication service providers. These authentication providers have total control over the user data to which they can leak or modify at their will. Thus OAuth 2.0, even though more secure from certain standpoints, raises serious security and privacy concerns from others [19].

When considering the above presented reading, we can conclude that even today there are many vulnerabilities to overcome and many improvements to be made in the security and authentication models of information systems. We believe to mitigate these vulnerabilities by utilizing blockchain technology and a blockchain based user authentication model. Table 3 provided below illustrates what our proposed platform offers in terms of security and user authentication, when compared to the solutions currently being used.

- **CEIMS** – Centralized Educational Institution Management System
- **ESMS** – School Management System for Government Schools in Ethiopia Using Distributed Database.
- **EICS** – Education-Industry Cooperative System Based on Blockchain
- **EduCTX** – A Blockchain Based Higher Education Credit Platform
- **HTTPBA** – Represents systems which incorporate HTTP basic authentication service.
- **HTTPDA** – Represents systems which incorporate HTTP digest authentication service.
- **OAuth 2.0** – Represents systems which incorporate OAuth 2.0 authentication service.
- **CreditX** – A Decentralized and Secure Credit Platform for Higher Education Institutes Based on Blockchain Technology (Solution proposed in this paper).

Table 3. Research gap 3

Features	HTTP BA	HTTP DA	OAUTH 2.0	EICS	EduCTX	CreditX
Encrypted credentials	✗	✓	✓	✓	✓	✓
Credentials need not be submitted with each request	✗	✓	✓	✓	✓	✓
Not vulnerable to man in the middle attacks	✗	✗	✓	✓	✓	✓
Not vulnerable to relay attacks	✗	✓	✓	✓	✓	✓
Not dependent on third party service providers	✓	✓	✗	✗	✗	✓
No privacy concerns regarding user data	✗	✗	✓	✓	✓	✓
Applied in a working product	✓	✓	✓	✗	✗	✓

2 Methodology

2.1 System Overview

The proposed system allows third party employers to verify their applicant's academic performance via an online interface while ensuring all the academic transcripts are securely stored in a private blockchain. Also, students are able to check their academic transcripts knowing that those are safe from attackers.

The proposed system is made up of four major components: a) The smart contract, b) The frontend module, c) User authentication module and d) The private blockchain. The smart contract is deployed into the private blockchain to startup the system. Then the frontend module which is hosted somewhere else is connected to the smart contract via JSON RPC calls, and finally the user authentication model will provide secure user authentication and will be incorporated by the frontend application. Furthermore, it will provide the ability for users to sign every transaction with their private key in a convenient manner.

2.2 Implementation

Smart Contract. The smart contract is where all the business logic is written. This is written using a programming language called [26] Solidity, which can be used to deploy in an Ethereum blockchain. This component also handles access control features such as, who can view or edit data, using custom access control modifiers which are written using user defined logic. An extra level of access control is provided in every function using conditionals, to ensure the integrity of the system. All the data must be stored in the blockchain, so every relation is stored as a mapping in the smart contract, which can only be accessed by authorized accounts. Calculation of the credits, a student has obtained for a module is done according to Table 4 considering the module offers four credits. When the number of offered credits varies, obtained credits will change according to the obtained mark and the number of offered credits.

Table 4. Credit allocation

Mark (α)	Grade	Credit
$\alpha \geq 90$	A+	4.0
$89 \geq \alpha \geq 80$	A	4.0
$79 \geq \alpha \geq 75$	A-	3.7
$74 \geq \alpha \geq 70$	B+	3.3
$69 \geq \alpha \geq 65$	B	3.0
$64 \geq \alpha \geq 60$	B-	2.7
$59 \geq \alpha \geq 55$	C+	2.3
$54 \geq \alpha \geq 45$	C	2.0
$44 \geq \alpha \geq 40$	C-	1.7
$39 \geq \alpha \geq 35$	D+	1.3
$34 \geq \alpha \geq 30$	D	1.0
$29 \geq \alpha \geq 0$	E	0.0

Frontend Module. The frontend module contains three vital subcomponents: 1) User interfaces (UIs), 2) Smart Contract Application Binary Interface (Smart Contract ABI) and 3) Frontend logic to interact with the blockchain. This component is mainly built using ReactJS, a frontend library which helps to build UIs, state management and routing between pages. Web3JS is a library which helps the frontend module to interact with the smart contract using JSON RPC calls. Those JSON RPC calls are defined in the smart contract ABI, which is generated using the actual, developed smart contract while compiling through Truffle. These JSON RPC calls are tightly coupled with the smart contract compared to a REST approach. When calling data in bulk from the smart contract, each item must be called out separately, so these types of functionality will have to be handled from the frontend module. The frontend module will also constantly monitor the activity of the smart contract to keep the UIs updated as much as possible.

User Authentication. The blockchain-based CreditX user authentication model mainly consists of two components. 1) The smart contract deployed in the Ethereum network, 2) The authentication server. The service allows users to sign into any platform which incorporates the authentication model. The authentication process also eliminates the need for the use of Web3 providers such as MetaMask plugin during the singing in process. It provides users with the convivence of using username, password combinations, which they are familiar with. A platform which wishes to incorporate the authentication service must expose a REST endpoint for the authentication process. Each account on the Ethereum blockchain is identified by an address which consists of 42 characters. These addresses are long and difficult for users to remember. The authentication model provides the ability for users to map their Ethereum address to a unique username on the blockchain. For each new user registering a private-public key pair is generated. During the registration, a message which contains the selected username and the users' public key is sent to the smart contract signed with the new users' Ethereum address. The authentication server, on the other hand, maintains users' private key and the hash of the password, which will be used for the authentication process. During the login, the authentication server verifies whether the username is registered in the blockchain. If registered it will allow users to enter the password. Once the password is entered, a token will be generated containing the hash of the hashed password plus a randomly generated string. This token alongside the randomly generated code will be sent to the REST endpoint, which in turn will generate a cypher encrypted with the user's public key. Finally, the cypher alongside the code and the token will be redirected to the verification endpoint of the authentication server. The server then will retrieve users' hashed password and the private key from the database. Where it will try to regenerate the same token by hashing the combination of the code plus the hashed password. The success of the authentication process will depend on this token validation process and the cypher being successfully decrypted by users' private key.

Private Blockchain. An Ethereum private blockchain will be used to deploy the smart contract of this system, which is developed using Geth (Go Ethereum) which is one of the original implementations of the Ethereum. This is heavily customized to complement the CreditX smart contract via configuration. This private blockchain is able to startup at one click and start the mining process, which is done via a GUI. In addition to that an external web app is provided to view the status of each node such as average block mining time, the last block mined time, etc.

3 Results and Discussion

The parameter settings of a private Ethereum network have a tremendous impact on performance. Since this product is mainly focused to build a secure, trustworthy credit platform, performance can be considered as a trading functionality. On a 2 core 4GB RAM computer, created private chain took 10 min and 2 s to mine its genesis block.

Smart Contract that represents Credit x's business logic is written as a single contract to reduce the gas expenditure. Also, this approach reduced the mining time of a block because the mining process took one address to sign the transaction. When working with

DAPPs and smart contracts, we found there were some limitations and programming barriers. To overcome those, we had to build some utility methods such as custom loops and configurations.

Access control modifiers used in the smart contract and in-build authentication server increased the secure nature of the CreditX platform. In the development process we managed to exclude the Ethereum accounts password from API calls. Furthermore, the user authentication model was able to provide highly secure and convenient user authentication.

The Credit X system is initially intended for the Sri Lanka Institute of Information Technology. In time, it can be scaled up to make a common platform for all the universities in Sri Lanka. It will make the whole system more robust, stable, and secure as the number of nodes increases. With the transparency of the Credit X, employers will be able to easily pick-up talent for their companies.

4 Conclusion

The CreditX system is initially intended as a proof of concept/work of the application of Blockchain technology in the education domain with the concentration of student credit information management in HEIs. However, it can be easily scaled up to make a common platform for all the universities in Sri Lanka. This will make the entire system more robust, stable, and secure as the number of nodes increases. Furthermore, with the functionality provided by the system, employers and other stakeholders will be able to easily acquire talent for their companies.

After scaling up to the national level, the system could be extended with additional features to replace the traditional student credit information systems in all higher educational institutes. Furthermore, since the concept is proven by the CreditX platform, the use of this technology and methodology can be easily adopted to support systems in other domains as well.

References

1. Turkovic, M., Holbl, M., Kosic, K., Hericko, M., Kamisalic, A.: EduCTX: a blockchain-based higher education credit platform. *IEEE Access* **6**, 5112–5127 (2018)
2. Reiff, N.: “Blockchain Explained”, 1 February 2020. <https://www.investopedia.com/terms/b/blockchain.asp>. Accessed 21 February 2020
3. Hooda, P.: “Comparison – Centralized, Decentralized and Distributed Systems”. <https://www.geeksforgeeks.org/comparison-centralized-decentralized-and-distributed-systems/>. Accessed 21 February 2020
4. Scott, D.A.: “Data Centralization-Advantages of centralized information system”, 11 July 2018. <https://www.confianzit.com/cit-blog/data-centralization-advantages-of-centralized-information-system/>. Accessed 21 February 2020
5. Whittaker, Z.: “Tufts expelled a student for grade hacking. She claims innocence,” TechCrunch, 8 March 2019. <https://techcrunch.com/2019/03/08/tufts-grade-hacking/>. Accessed 21 February 2020
6. Kausar, E., Islam, M., Ahmed, T.: “Centralized Educational Institution Management System”. <http://dspace.bracu.ac.bd/xmlui/handle/10361/9026>. Accessed 20 February 2020

7. Amare, S., et al.: “Centralized School Management System for Government Schools in Ethiopia using Distributed Database” . <http://ijettjournal.org/2018/volume-60/number-2/>. Accessed 19 February 2020
8. Gilbert, S., Lynch, N.: Perspective on the cap theorem. Computer **45**(2), 30–36 (2012). Accessed 20 February 2020
9. Media Lab Learning Initiative. Digital Certificates Project. <http://certificates.media.mit.edu/>. Accessed 15 February 2020
10. Universidad Nacional De la Plata. <https://www.unlp.edu.ar/>. Accessed 15 February 2020
11. Amati, F.: First Official Career Diplomas on Bitcoin’s Blockchain (2015). <https://blog.sig-natura.co/first-official-careerdiplomas-on-bitcoins-blockchain-69311acb544d>. Accessed 16 February 2020
12. “Sony Global Education”. <https://www.sonyged.com/ja/>. Accessed 20 February 2020
13. Al Harthy, K., Al Shuhaimi, F., Al Ismaily, K.K.J.: The upcoming Blockchain adoption in Higher-education: requirements and process. In: Conference: 2019 4th MEC International Conference on Big Data and Smart City (ICBDSC), IEEE Access (2019)
14. European Commission, “European Credit Transfer and Accumulation System (ECTS)” . https://ec.europa.eu/education/resources-and-tools/european-credit-transfer-and-accumulation-system-ects_en. Accessed 21 February 2020
15. Alammary, A., Alhazmi, S., Almasri, M., Gillani, S.: Blockchainbased applications in education: a systematic review. Appl. Sci. **9**(12), 2400 (2019)
16. Liu, Q., Guan, Q., Yang, X., Zhu, H., Green, G., Yin, S.: Education-Industry Cooperative System Based on Blockchain. In: 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN) (2018)
17. Mertz, L.: (Block) chain reaction: a blockchain revolution sweeps into health care, offering the possibility for a much-needed data solution. IEEE Pulse **9**(3), 4–7 (2018)
18. Shakbulatov, D., Arora, A., Dong, Z., Rojas-Cessa, R.: Blockchain implementation for analysis of carbon footprint across food supply chain. In: 2019 IEEE International Conference on Blockchain (Blockchain) (2019)
19. Patel, S., Sahoo, A., Mohanta, B.K., Panda, S.S., Jena, D.: DAAuth: a decentralized web authentication system using Ethereum based blockchain. In: 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN) (2019)
20. InformIT, “HTTP Basic Authentication”, 12 February 2002. <https://searchwindowserver.techtarget.com/tip/HTTP-basic-authentication>. Accessed 19 February 2020
21. Arslan, E.: “Implementation of MD5 Cryptographic Hashing: Digest Authentication”, 31 December 2017. <https://medium.com/@eyupcanarslan/implementation-of-md5-cryptographic-hashing-digest-authentication-c23bd65eef07>. Accessed 17 February 2020
22. Dan Arias, “Hashing in Action: Understanding bcrypt”, 31 May 2018. <https://auth0.com/blog/hashing-in-action-understanding-bcrypt/>. Accessed 20 February 2020
23. Pecanac, V.: The HTTP series (Part 4): Authentication Mechanisms”, 17 July 2017. <https://code-maze.com/http-series-part-4/>. Accessed 16 February 2020
24. Anicas, M.: “An Introduction to OAuth 2”, 21 July 2014. <https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>. Accessed 18 February 2020
25. Cardoza, C.A.: “The importance of OAuth 2.0”, 8 December 2017. <https://sdtimes.com/app-security/importance-oauth-2-0/>. Accessed 20 February 2020
26. Frankenfield, J.: “Smart Contracts: What You Need to Know,” 8 October 2019. <https://www.investopedia.com/terms/s/smart-contracts.asp>. Accessed 15 July 2020



An Architecture for Blockchain-Based Cloud Banking

Thuat Do^(✉)

Department of Mathematics, Hong Kong University of Science and Technology,
Clear Water Bay, New Territories, Hong Kong

Abstract. Blockchain has been practiced in crypto-currencies and cross-border banking settlement. However, no clear evidence that a distributed ledger network (or Blockchain) is built within domestic payment systems, although many experts believe that Blockchain has wide applicability in various industries and disciplines. As the author's best knowledge, no one has published a clear architecture and a feasible framework for a Blockchain-based banking network. Thus, "*how Blockchain can be implemented in domestic banking systems*" is a big challenge. The most important contribution of this work is to give a feasible and viable framework resolving that problem. The author investigates a Blockchain-based payment framework, more explicitly, a decentralized banking architecture running on the top of existing banking cores. The Blockchain network has two tiers: master nodes (block generators) and normal nodes (validators). The consensus mechanism is introduced as a composition of Proof of Stake, Proof of Reputation and/or practical Byzantine Fault Tolerance. In addition, nomination and approval mechanisms are added to the governance to enhance legal compliance and compatibility with real Fintech space. Some qualitative analysis is provided to show that the proposed Blockchain banking framework offers better security, scalability and decentralization, while easily adapt with different national regulation environments, among other Blockchains. In the application aspects, the framework is implementable and deployable for decentralized payment network and smartcontract infrastructure for domestic markets, then enable a complete and unified digitized space for cloud banking and financial services.

Keywords: Blockchain · Byzantine fault tolerance · Cloud banking · Decentralization · Distributed ledger · Distributed ledger technology · Proof of stake · Proof of reputation

1 An Introduction

1.1 A Literature Review on Blockchain and Its Applications

Distributed Ledger Technologies (DLTs) are based on two fundamentals: cryptography (public keys, hash functions) and consensus mechanism. Its goal is

to create a unified and trusted ledger which is secure, always available, shared among the involved parties and impossible to control by any single party. In terms of information technology, a distributed ledger is simply a replicated database of transaction data and some other information (e.g. coin reward, messages).

Based on architecture, all the implemented DLTs can be classified into three types: Blockchain, Tangle and Hashgraph. While the two laters are complex systems based on the Directed Acyclic Graph structure, the former is easily understood as its name, *a chain of blocks*. IOTA is the most famous project deployed Tangle [17] since 2017. Hedera [7] is a typical project applied Hashgraph since 2019. However, Blockchain is the most popular, intensively studied and developed DLT which is original from Bitcoin and proven via many notable projects, for instances, [Ethereum](#), [Ripple](#), [Corda-R3](#), [Azure](#) (Microsoft), [Quorum](#) (acquired by Consensys from JP Morgan Chase).

Blockchain has become a new HOT industry worldwide not only in cryptocurrency communities but also among scientists, technologists, developers and regulators. Chinese government has promoted Blockchain as a breakthrough technology and gave huge support for research and development, targeting the leading position of the nation in the new space. According to CBINSIGHTS's report [8], Blockchain and DLTs can revolutionize the global financial sector worth around 134 trillion dollars, ranging in the following zones.

Payment. By establishing a distributed ledger, Blockchain provides faster payment with lower cost in comparison to current banking systems. Cross-border payment is usually complicated, time-consuming and costing 5–20% remitted amount, while Blockchain is believed to cut down the fee to 2–3%.

[BitPesa](#) is a B2B payment Blockchain-based company operating in Kenya, Nigeria and Uganda, gained over 25,000 customers after 5 years, processed more than 1 million transactions worth of 340 million dollars. [BitPay](#) is another Blockchain-based payment company in the US, funded 72 million USD, accepting bitcoin payment.

Clearing and Settlement. [Ripple](#), a Blockchain startup specializing in bank settlement, estimates that it could cut 33% fee compared to SWIFT. Ripple has more than 100 customers. [Stella](#) collaborates with IBM to develop a Blockchain-based international payment for 44 banks in over 72 countries with 47 currencies. It only takes a few seconds to complete a transaction on its Blockchain. [Corda-R3](#), a distributed ledger platform for bank settlement, aims to become a new *operating system* for the financial market. It has raised 107 million dollars from Bank of America, Meryll Lynch and HSBC in 2017.

Identity Verification. This crucial process normally requires many steps and takes long time, multiple duplicated among financial institutions and companies. Blockchain helps create a decentralized, easily accessible, fast verifiable

and secure database of digital identities with privacy. Cambridge Blockchain and [Tradle](#) are fintech startups which utilize Blockchain to enhance various procedures in banks with the help of a customer verification system.

Security Token Offering and Digital Asset Exchange. In 2015, Nasdaq planned to use Blockchain for their private market platform, with the introduction of [Colored Coin](#) concept, to distinguish coins used in transactions with other types. Nasdaq joined Citigroup to invest in [Chain](#), a Blockchain company which provides a reliable decentralized database that records all stock and ownership transactions in real time. Putting stock on Blockchain could save 17 to 24 billion dollars annually for global processing fee.

Credit and Syndicated Loan. By eliminating the *gatekeeper* in lending and credit, Blockchain can reduce risk. In 2016, Credit Suisse, Symbiont, R3 and Ipreo completed the first stage of a project using Blockchain in syndicated loan market. In April 2018, international banks, BNP Paribas, BNY Mellon, HSBC, ING, Natixis and State Street, jointly supported Fusion LenderComm by Finastra, a Blockchain platform for syndicated loan. BBVA, Mitsubishi UFJ and BNP Paribas gave a 150 million dollars of syndicated loan to Red Electrica, a Spanish electronic company. The [event](#) was recorded on Ethereum.

Trade Finance. By simplifying procedures, Blockchain can strengthen transparency, security and trust among partners on the globe. [TradeLens](#) (Maersk and IBM joint venture) and [eTradeConnect](#) (formed by Hong Kong banks) are notable distributed platforms for trade finance. [Voltron](#) (under R3 and Crypto-BLK) operates Blockchain-based platform for letter of credit application.

Crowdfunding. Initial Coin Offering (ICO) is a new way for tech startups to approach funding. In the first half of 2018, ICOs raised 13.7 billions dollars, doubled from 7 billion in 2017, according to Businessinsider [9].

Accounting and Audit. Distributed technology can help remove lots of paperwork involved in this field. Blockchain can become as a decentralized notary. Furthermore, smart contracts are useful for automatic invoicing. PricewaterhouseCoopers has developed Blockchain-based accounting service for enterprises.

According to Gartner [6], Blockchain is one of the most promising technology trends and can generate more than 176 billion USD by 2025, and 3.1 trillion USD by 2030. The technology is not yet mature, but it is temporary. Blockchain has been studying and developing extensively with significant progress.

1.2 The Paper's Structural Contents

The article consists of seven sections. The first one gives a brief introduction about Blockchain (more generally, distributed ledger technologies) and its appli-

cations in real world. The second one presents the status and challenges of implementing the technologies in banking sector, the inspiration to build a feasible framework of Blockchain cloud banking. Section 3 describes the framework in details, for instances, network architecture, workflow, core protocols. The next section shows a deep discussion on governance of the Blockchain-based banking network, then it introduces node management mechanism and reputation system over the network. Section 5 presents block production and consensus process. The consensus is a modification of Delegated Proof of Reputation (DPoR) introduced in [18], and can be viewed as a hybrid of Proof of Stake and a reputation ranking system. Section 6 differentiates the proposed framework with existing Blockchain networks (both public and private/enterprise), then analyzes its advantages. The final section gives application perspectives, assessment and conclusion. The most important contribution of the paper is a Blockchain framework for cloud banking with network architecture, governance and consensus mechanisms clearly described.

2 Blockchain in Domestic Banking: Challenges and Inspiration

When studying applications of Blockchain in banking and finance, people immediately think about global crypto currencies (e.g. bitcoin, ether, Libra, etc.), international settlement or money transfer (e.g. Ripple, Corda-R3, Quorum which aim to replace Swift Code protocol), international trade-finance (TradeLens, eTradeConnect). Obviously, there are big barriers in cross-border value transfer that Blockchain can erase. Nonetheless, “*how Blockchain can disrupt national banking systems? Can it renovate a national payment gateway, i.e. interbank payment?*” are big questions without any example or proposal out there. Relating Central Bank Digital Currency (CBDC), big organizations has published investigations and reports combining both regulation and technology consideration. World Economic Forum (WEF) indicated 10 use cases that central banks can apply DLT [1]. After that WEF provides a study and assessment toolkit for CBDC policy maker [3], which mentions stablecoins as an alternative and example for CBDC. Bank for International Settlements (BIS) appreciates DLT to offer a resilient digital currency system [2]. Brookings [12] presents design choices for CBDC together with deep investigations on centralized and decentralized ledgers, digital identification, digital wallet, account and UTXO models. China has developing its digital Yuan (already in piloting stage) but no backed peer-to-peer network or Blockchain design behind the digital currency is disclosed or open apparently (read more in [19]). Overall, developers cannot find any implementable framework in the mentioned studies to build a Blockchain (or DLT) network for domestic banking applications.

Alipay, Wechatpay, M-Pesa have been successful to provide a frictionless and seamless payment, even more extensive banking and financial services (e.g. saving, investment) to end-users everywhere simply via mobile, without coming to bank offices. Based on connections with bundles of banks (via APIs), they

provide many banking services, but they are not truly banking institutions with full regulatory compliance (which is unfair for banks). In addition, they are centralized escrows possibly causing concerns on monopoly, financial security (single failure), transparency and privacy. No one can find a *ready-to-run* open API platform for banking services (i.e. Inter-Banking Cloud Infrastructure as a Service), which allow many fintech firms to provide inclusive banking and finance applications to end-users seamlessly with low implementation cost and without friction.

Smartcontract is successfully applied in crypto space but not in the conventional industries, although its wonderful potential of applications in various disciplines and landscapes is described extensively. For example, smartcontract can function as the second-layer of a CBDC system and boost innovation from commercial banks and fintech developers (Section 7, [2]).

Although many researches on application of Blockchain and DLTs in banking sector have been conducted, there is no feasible framework for implementation. Therefore, the author is going to design a Blockchain network architecture for banking and financial industries, attaching with core protocols, governance and consensus mechanisms. In the next sections, the terms of the Blockchain, the network, the cloud, the banking cloud, the Blockchain banking cloud all refer to our proposed framework (i.e. the Blockchain-based banking cloud, unless otherwise specified. Analogously, the terms of account, wallet, address are used alternatively, unless otherwise specified. Readers can understand that *banking cloud* refers to a decentralized IT infrastructure while *cloud banking* means banking services running on the cloud. However, in this paper, those two terms can be used alternatively.

3 A Framework for Blockchain-Based Banking Cloud

3.1 Sketching a Network Architecture

The author envisions a Blockchain-based banking network, in other word, a decentralized infrastructure for various banking and financial applications, e.g. money transfer, payment, saving, investment, etc. The network can function a decentralized cloud banking infrastructure as a service, running on the top of existing core-banking systems. Then all banking and financial services can be implemented as decentralized applications, running on the top of the cloud (i.e. the Blockchain), which utilize all advantages of Blockchain and smartcontract, while preserving the essential legal compliance and security of the banking systems.

3.2 Two-Tier Network and Workflow

The proposed network is not pure peer-to-peer like Bitcoin, Ethereum and other public Blockchains. It consists of two classes (or tiers): master nodes and normal nodes (see Fig. 1) with different right and role. The necessary and sufficient conditions to become such a node will be presented in Sect. 4.

- **Master nodes** are block generators who hold the ultimate power, be able to function all core protocols and the consensus mechanism, and store full block-data. Only legitimate banks can become master nodes. The block generators verify submitted transactions, then gather in a block and finalize it according to the consensus mechanism (presented in Sect. 5).
- **Normal nodes** are transaction validators. Banks, financial institutions, payment service providers, big merchants, can join the network as normal nodes. The validators receive transaction proposals from clients (i.e. from end-users), validate them and then forward to master nodes for confirmation and finalization.

In general, clients may submit their transaction proposals directly to master nodes. However, master nodes give priority to the validated transaction pool. Normal nodes help validate transactions before sending to master nodes, thus reduce the block generators' workload. The workflow is provided in Fig. 2. In addition, the network allows nodes attaching their private chains (or private payment channels) on the main chain, thus improving the overall scalability and performance.

3.3 Block-Data and Distribution

Since the network is classified into two distinguished tiers, its block data should be designed in a different way to assure appropriate compliance and privacy. Block data is divided into three parts: Header, State and Body (see Fig. 3). A short description for block data attributes is given below.

- The **Header**
 1. Hash of previous block (i.e. hashing value of the previous block data).
 2. Time stamp presents the time point of block generation.
 3. Root hash of the Merkle tree.
- The **State**
 1. New registered identification hashes (e.g. hashing values of [*name, identity number, birthday*]).
 2. New registered account addresses (e.g. hashing[*bank code*] + hashing[*public key*]).
 3. Balance and state updates show new state changes on the entire network, (e.g. [*identity-account mapping, account addresses, available balance, other new states*]).
 4. Coinbase presents rewards (paid via a native token like bitcoin, ether). Feebase describes stable-coin-based transaction fee. Rewards and fees are accompanied with an appropriate distribution over the nodes.
- The **Body**
 1. Clearing updates show clearing statistics among participant banks (commonly master nodes only) so they can proceed to settlement, e.g. via an outside clearing house.

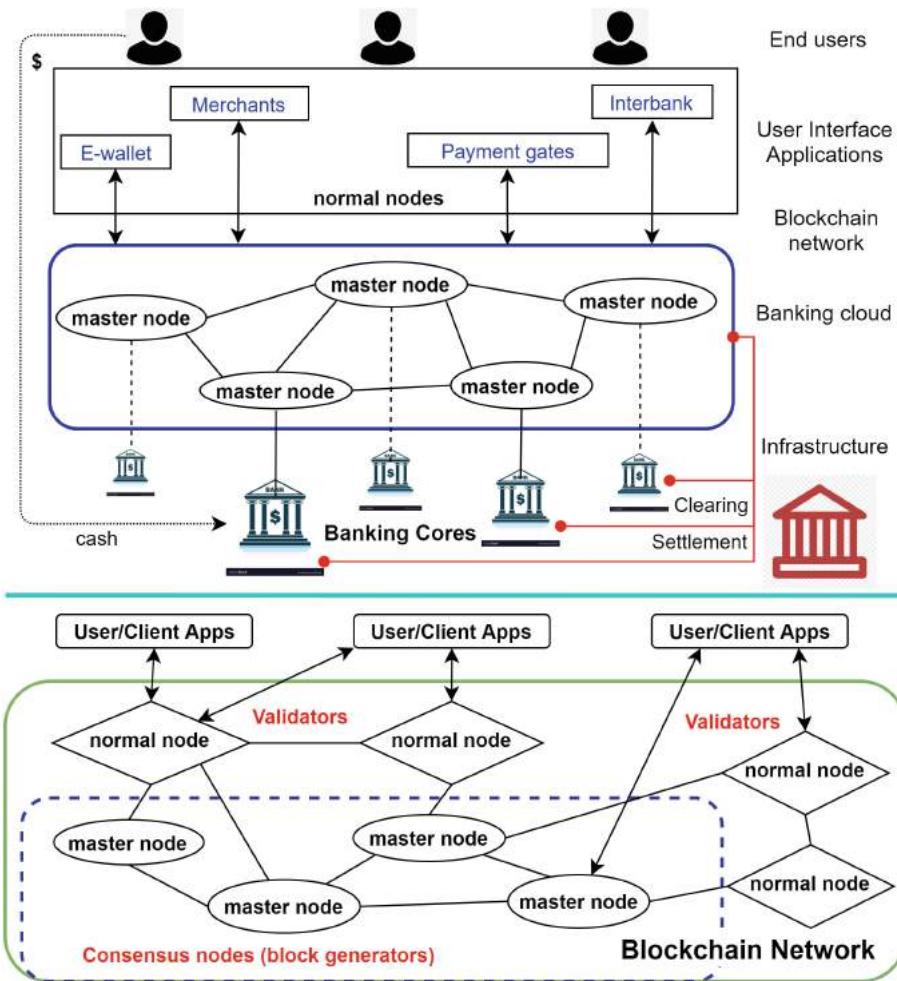


Fig. 1. Sketching the blockchain banking cloud: block generators and transaction validators.

2. Transaction records show detail information of all transactions and their hash values, e.g. [sender's addresses, receiver's addresses, number of transferred tokens].

The proposed block data and its distribution differ with existing public Blockchains in the following major points (also read more in Sect. 3.4).

- **Account and identity:** Identification hash is mapped with a real identity stored off chain. An identification hash may be attached with (at least) one or many account addresses. Every account must be mapped with at least one identity.

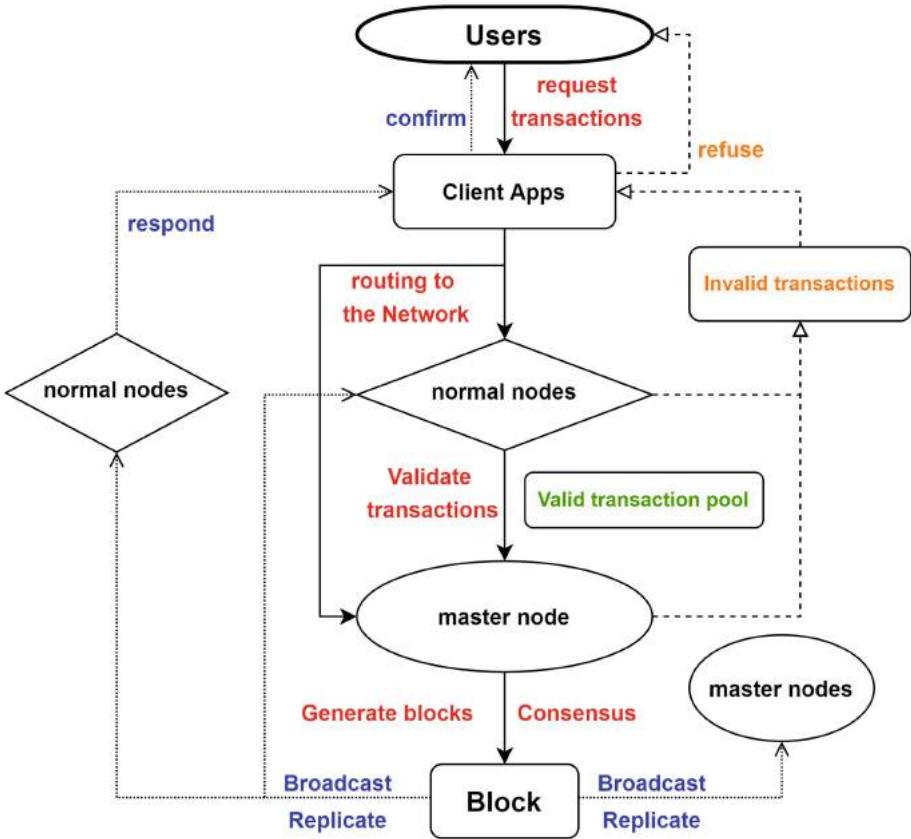


Fig. 2. A basic workflow.

- **Coinbase and feebase** presents a dual-token model. The coinbase utilizes a native token (like bitcoin, ether) to incentivize participating nodes for contribution to the network operation. The feebase uses stable coins (i.e. digitized fiat currencies based on bank pledge) as transaction fee utilities, except native coin transactions.
- **Clearing update** gives clearing statistics among participant banks so they can proceed to settlement, especially in real time without a centralized clearing house, provided a builtin central bank digital currency.

Master nodes play the critical role of full block data storage and full operation on the network. When a block is produced and endorsed, the generator will broadcast it fully to other master nodes, the header and the state to normal nodes. The normal nodes can use the header and the body for Simplified Payment Verification (SPV) and transaction validation without asking master nodes. SPV nodes on Bitcoin Network must ask full nodes to verify certain transactions. Thus, normal nodes in our network are neither precisely equivalent to

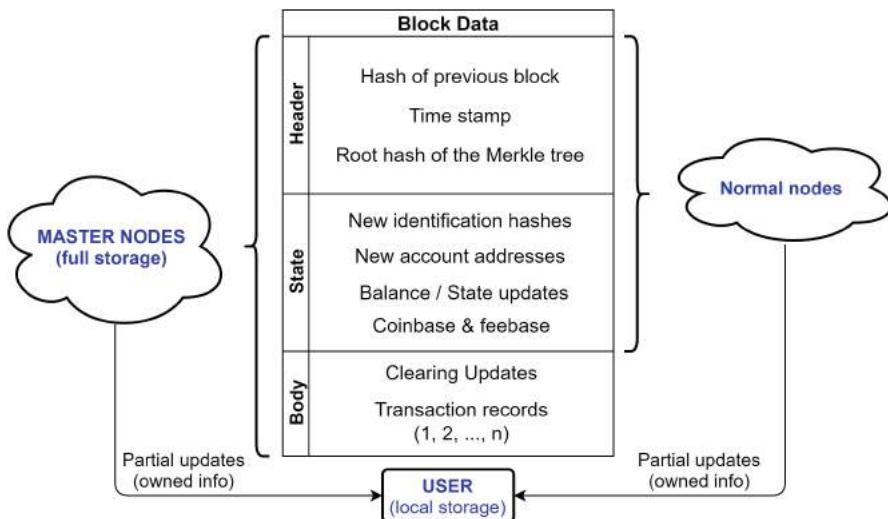


Fig. 3. Block data is distributed differently among participants.

Simplified Payment Verification nodes nor full nodes on Bitcoin or Ethereum. In addition, block data (specified relevant transaction info) is partially updated to the associated users. This task is normally done by normal nodes and client servers.

3.4 Core Protocols

We are going to describe the core protocols of the proposed Blockchain-based banking cloud, which are identification protocol, incentive protocol, stable coin protocol and clearing protocol (the three laters are for master nodes only).

Identification Protocol utilizes hash (checksum) techniques to help identity verification better and faster. The Blockchain network does not store customer identity information but its hashed value for cross verification. Member banks and other network participants keep their own customer identity information in their own private database. The only thing the members do is registering (i.e. submitting) identity hash values on the network. Note that each identity hash is identical with one and only one body (e.g. an individual, a company or an organization).

The procedure is as followed.

1. An user request opening an account on the cloud banking network. The node storing his identity info will hash the data, then broadcast the hashed value associated with (at least) one or many new public addresses to the

network. The registration is complete once the identity hash and its associated address(es) are included in a confirmed block. Then the user can make transactions.

2. If no identity information exists, then the user is required to complete Know Your Customer (KYC) process. After KYC, it returns Step 1.

The identification hash helps the network's participants verify the existence of an identity while keeping the original identity information confidentially outside the Blockchain. This protects privacy and confidentiality while facilitating cross verification, certification and information exchange. Note that all public Blockchains (e.g. Bitcoin, Ethereum) store anonymous addresses without any mapping to real identities. Our proposed Blockchain is anonymous on chain but every account is associated with a verified KYC info stored off-chain (at least in some node's private database).

Incentive Protocol is implemented at the bottom of the Blockchain, and only master nodes (as block generators) have the right to function and maintain it. The protocol issues a unique native token (or native coin) utilized for staking (in the consensus mechanism) and rewarding on the entire Blockchain network. The native coin represents the intrinsic value of the Blockchain cloud banking network analogously to crypto assets (e.g. BTC of Bitcoin, ETH of Ethereum), which may varies over time. Therefore, no stable coin (e.g. digitized fiat types) can satisfy that special nature. The protocol will pre-mine a certain amount of native coins at the genesis block (i.e. block 0) to use for initial staking of the foundation nodes. After that the new coins are generated as reward per newly produced block and distributed appropriately to all nodes, and possibly to the development foundation. This is clearly indicated in the coinbase of the block state (see Sect. 3.3). For example, the parameters of the incentive protocol can be set as followed (also read Sect. 4).

- The block reward rate is max 2% of the current supply per year, evenly divided per block. One can set a maximal supply (e.g. 1 billion native coins), i.e. the protocol will not generate new coins any more after reaching that number.
- In each block, coinbase protocol computes and distributes new generated coins and transaction fees (applied for native coin transactions and specified executions only) as the following. Assuming there is N nodes on the Blockchain network.
 - 10% or $1/N$ (for which smaller) to the block generator.
 - 5% or $1/2N$ (for which smaller) to the block endorsers who co-sign to finalize the block (if any), other than the generator.
 - 10% or $1/N$ (for which smaller) to the transaction validators (evenly divided by the number of native coin transactions included in the block),
 - 3% or $1/3N$ (for which smaller) to the development foundation.
 - The rest 72% will be distributed evenly as 44% to all master nodes and 28% to all normal nodes.

Stable Coin Protocol (more explicitly the fiat digitization protocol) is implemented at the bottom of the Blockchain, i.e. only master nodes have the right to operate and maintain it. The protocol allows issuing digital tokens 1-to-1 corresponding to equivalent cash pledged in bank and burning the tokens corresponding to cash withdrawal amount. This means no new currency is issued. The token is simply an accounting representative of cash reserve in bank. The protocol also provides digitized fiat balance update of all accounts on the network. Zero Knowledge Proof algorithms can be implemented to blind user balance updates (in the state of block data) sending to normal nodes. An implementation of shielded transactions and addresses can be found on Tron using zk-SNARK [4]. The purpose is protecting privacy on the entire network while allowing easy verification and fully tracking on the master nodes and the account holders, respectively.

The stable coin is issued based on user request, and the cashin and cashout procedures are as followed.

- **Cashin** allows users (with registered account on the network) deposit to the Blockchain network based on their cash balance in bank. For example, an user requests a deposit of \$1000, his home bank verifies and confirms if his cash balance is enough. Then the bank (also a master node) issues \$1000 stable coins to the user's Blockchain account.
- **Cashout** allows users withdraw cash from his own balance on the Blockchain. For example, an user requests a withdrawal of \$1000, any member can verifies his balance and confirms cash providing. Then \$1000 stable coins are deducted from the user account and sent to the burning address (containing exactly non-reusable redeemed stable coins).
- Cashin must be executed by associated banks while cashout can be provided by any member of the network.
- Fee for Cashin and Cashout is cash basis and depends on the service providers.

In addition, the stable coin protocol allows nodes to setup their desire fee for transaction validation and confirmation, except native coin transactions. The protocol also returns feebase to be included in the state of block data (Fig. 3). In addition, the protocol allows multiple stable coin issuance. Each type of stable coins can be easily converted to others via atomic swap techniques which has many practiced implementations in crypto-currency space.

Clearing Protocol is implemented at the bottom of the Blockchain, and only master nodes have the right to operate it. Per block, the protocol reads

```
#per participant bank, run
    Cash_flow = Cashin - Cashout
#then for the entire network, return
    Clear(positive Cash_flow group, negative Cash_flow group)
```

The *Clear* function returns the outside clearing house a statistics for further settlement and state update within the corresponding banks. Note that the following equation is always hold per block

$$\sqrt{cashin} = \sqrt{cashout + \sqrt{fees + \sqrt{stable_coins}}}. \quad (1)$$

Note that in Eq. (1), the fees (if applied) are paid in stable coins for all transactions, except native coin transactions, and the stable coins are usable (i.e. not redeemed).

Unlikely pairwise clearing method used in central clearing houses, our Blockchain banking cloud enables group clearing among multiple parties. For example, assuming that there are five banks $\{B_1, B_2, B_3, B_4, B_5\}$ in the clearing process.

The central counterparty always does all pair-wise clearing and settlement (10 computations and 10 updates) per transaction update

```
clear(B_1,B_2)  clear(B_2,B_3)  clear(B_3,B_4)  clear(B_4,B_5).
clear(B_1,B_3)  clear(B_2,B_4)  clear(B_3,B_5)
clear(B_1,B_4)  clear(B_2,B_5)
clear(B_1,B_5)
```

On the other side, our clearing protocol only does clearing and settlement between the positive and negative groups, hence simplifies and speeds up the process. Moreover, it offers a capability of real time settlement provided a builtin CBDC. For example, without loss of generality, assuming that the banks has the corresponding balances, $\{+x_1, +x_2, +x_3, -x_4, -x_5\}$, where $x_i > 0, i = 1, \dots, 5$. Then the protocol does 5 computations and 5 updates per block

```
Y = x_1 + x_2 + x_3
Z = x_4 + x_5
Deduct (x_1/Y)*Z from B_1
Deduct (x_2/Y)*Z from B_2
Deduct (x_3/Y)*Z from B_3
Settle x_4 for B_4
Settle x_5 for B_5.
```

4 Governance on the Network

Our proposed Blockchain is neither permissionless nor permissioned. In fact, no central authority exists on the network. Then a nomination mechanism (see Fig. 4) is introduced to ensure that only qualified candidates have opportunity to join the group of network operators while preventing corruption caused by one centralized authority and potential malicious guys.

4.1 The Conditions to Become a Node

The Necessary Condition. To become an operating node, a candidate must stake a required minimum amount of native coins to activate (or register) the pair of public-private keys with the network. A higher minimum stake is required for master node role.

The Sufficient Condition. A body wishing to become a network operator (i.e. a master or normal node) must complete registration first, and then nomination. Explicitly, a candidate for normal node is required at least two nominations from normal nodes or one nomination from master nodes. A candidate for master node is required at least two master node nominators. One may ask an additional condition that the two nominators must possess (in summation) a minimum percentage (e.g. 10% or $2/N$ for which smaller) of the total reputation scores of all nodes, where N is the number of nodes. See reputation score in Sect. 2. Of course, at the genesis block, no nomination happens, and the foundation nodes are all promoted up by the developer legitimately. Note that, if the developer is not a legal licensed bank, it may not be a master node, because other foundation banks do not allow to build up such a network.

Node Deactivation. An active node can decide itself to leave the network, simply inform its leaving with other nodes. Another way, operating nodes can vote to kick out some node if it is not honest with proven damage evidences for the network, or it is not qualified longer under the assessment of the majority. Votes are weighted by reputation scores (see Sect. 2) and at least 68% of weighted majority from the network is needed to deactivate a node. This mechanism helps removing proven malicious nodes from the network and promoting high quality nomination, honest commitment and frequent activities.

4.2 Reputation Score System

Reputation is very important in social reality. It bases on two major factors: wealth and performance (or achievement). Thuat Do et al. investigated reputation in Blockchain space and introduced a novel framework for consensus, namely Delegated Proof of Reputation, see [18]. Therein, ranking (inspired by Google's PageRanking) is an essential component contributing to reputation, applied not only for nodes but also all accounts on the network. It exploits the idea that a node's ranking is built up over time based on its cooperation (connection) and work achievement. Basically, more valuable transactions, more connection will get higher ranking. This paper doesn't study the ranking algorithms. The author only takes the idea that ranking is helpful to reduce the number of faulty nodes and malicious actions on the network while promoting integrity and quality contribution, hence improve overall security and reliability on the network.

Removing resource power suggested by EOS [5] and [18], the reputation engine in this paper computes staking and ranking factors to return normalized

reputation scores. It is assumed that all qualified nodes have abundant resources to solve the tasks on Blockchain. The reputation scores are used for voting and choosing block generators, and formulated

$$Rep = \mu S + (1 - \mu)R, \quad (2)$$

where Rep is reputation score, S is normalized staking index and R is normalized ranking score. Readers refer to [18] for more detail on the reputation and ranking framework, computation and advantage analysis. The parameter μ in Eq. (2) is the control multiplier balancing staking and ranking components. In the early stage of the Blockchain, the numbers of connections and transactions are small (i.e. insignificant), thus μ should be large and then gradually decrease. One can set $\mu = \max\{2^{-h/K}, \theta\}$, where $0 < \theta \in 0.5$ is constant, K is a positive integer and h is block height. That means $\mu = 1$ at the genesis block, then monotonously decreasing to 0.5 at block K -th, and $\mu = \theta$ whenever $2^{-h/K} \in \theta$.

Reputation Penalty. If a node is kicked off from the network (see Sect. 4.1), then its nominators' reputation score will be considered as zero for 30 following days, although having positive values. Other punishments on reputation score can be voted and decided by the majority, then applied where appropriate.

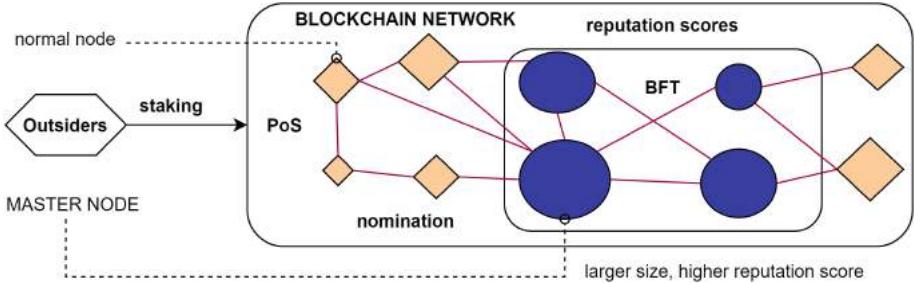


Fig. 4. Governance and consensus.

5 Block Production and Consensus Mechanism

A bundle of consensus protocols are proposed and implemented in various Blockchain networks. The most popular ones (by order) are Proof of Work (PoW), Proof of Stake (PoS) and Delegated Proof of Stake (DPoS). In this part, the author exploits Delegated Proof of Reputation (DPoR), introduced in [18], with a modification (i.e. removing resource power in overall reputation).

Accounts on the network are allowed to grant their reputation scores to operating nodes (this is similar to voting procedure in Tron [4] and EOS [5]). A node's reputation score (included granted quantities) is converted to the probability of the node to be selected as transaction validators, block generators, reputation

score providers and random source generators. Higher reputation score implies greater probability and hence earning more rewards and fees. Assume that the groups of master nodes and normal nodes are $\{M_1, \dots, M_p\}$, $\{N_1, \dots, N_q\}$ associated with reputation scores $\{Rep_i^M, \dots, Rep_p^M\}$, $\{Rep_i^N, \dots, Rep_q^N\}$, respectively, where p, q are positive integers. Then the normalized probability outputs are

$$Prob_i^M = \frac{Rep_i^M}{\sum_1^p Rep_j^M}, \quad Prob_i^N = \frac{Rep_i^N}{\sum_1^q Rep_j^N}. \quad (3)$$

Block producing process is divided into *epoches* and *gaps*. Each epoch contains a fixed length of blocks. Between two consecutive epoches, there is a short gap for conclusion on reputation score update and re-generating random sources. Based on the computed probabilities, the random sources determine:

- a reputation score result provider for the next epoch (or next K epoches);
- a random source generator for the next epoch;
- an ordered group of transaction validators among normal nodes;
- an ordered group of block generators among master nodes.
- an ordered group of gap-block generators (among all nodes) for the next gap.

To finish a certain gap, a special block (*gap-block*) is generated to record the info. A compensation (paid in native coins by the other nodes or coinbase reward) is given to the providers and the gap-block generator. The gap block (containing new reputation scores and random sources only) is valid and confirmed if at least 51% of the total reputation scores (including the granted quantities) of all nodes signed “*agree*”. In addition, the gap block refers to the previous one to make it a check-point for reference or recovery if necessary.

Basically, a block generator, in its turn, will choose transactions validated by legitimate validators to package into a block. After that, it broadcasts the block randomly to other master nodes for finalization via a practical Byzantine Fault Tolerance (BFT) algorithm. There are many practical BFT algorithms out there, see [10, 11]. An open source of BFT implementation is Tendermint Core [14] which has been deployed on Binance Chain [16]. Normally, BFT requires at least $f + 1$ replicas, $2f + 1$ nodes and $\mathcal{O}(n^2 f)$ communication complexity to ensure error-free process and fault tolerance system, given f faulty nodes. Thanking to nomination mechanism, it is expected to reduce f remarkably, then speed up communication and finalization process. BFT allows a secure and instant finality on blocks and transactions but its broadcasting process is complex and costs long time. If real-time finality is not a strict requirement, the author suggests replacing BFT broadcast by the longest chain rule (as Bitcoin, Ethereum, Tron applied). In fact, a transaction on Tron Network can be considered as final (or immutable) if there are 20 block confirmations (equivalently 60 s), not a long wait.

The Algorand [15] Blockchain uses a Byzantine Agreement protocol and verifiable random function in its so-called *pure PoS* consensus system. Such a random function is a usefully practical implementation to deploy on other PoS-based Blockchains, as random sources play a critical role in the selection of block generators and transaction validators fairly and honestly.

6 Differentiation and Advantage Offering

Our proposed Blockchain framework differs from all the existing public, private and consortium Blockchains in several beneficial ways.

Creative and flexible architecture with two tiers offers a high adaptivity and compatibility with various banking and IT systems, while allowing nodes to attach their private chains easily. Some public Blockchains (e.g. EOS, Tron) designate network participants into super nodes (block generators) and standby nodes (doing nothing). On our proposed Blockchain, every node has its own role and tasks.

Novel block data and heterogeneous distribution make the framework more friendly with privacy and confidentiality while fully ensuring necessary tracing and tracking. This improves the compliance and adoption among regulators and banking institutions.

Novel nomination mechanism guarantees stable network expansion with qualifying entrance, hence enhances reliability among nodes.

Reputation system offers better decentralization (via granting reputation score) rather than pure staking and voting mechanisms on Tron and EOS, while promotes nodes and application developers working honestly and actively to gain reputation (see more rationales in [18]). Note that PoW (resp. PoS) Blockchains have been facing centralization on giant mining pools (resp. staking concentration on capital whales).

The advantages of the Blockchain cloud banking can be easily pointed out, both on technology side and application perspectives.

Better Security, Scalability and Decentralization. Nomination entrance reduces potential faulty nodes, hence improve overall security. Two-tiers architecture with the capability of connecting private chains allows network scaling greatly. Reputation system promotes decentralization of network and builds trust on the network.

Fairness for Developers. Application developers are important value contributors of any Blockchain network. Unfortunately, they do not have any right in the existing Blockchains' governance, although their billion dollar business are running on their tops. Our Blockchain framework gives the developers a chance to join network operators based on their reputation.

Regulation and Adoption Friendliness. Banks are more pleasant to join the network because the master node design precisely presents their special role, right and responsibility in the banking and financial sector. KYC problem, privacy and confidentiality are all resolved by the identification protocol (see Sect. 3.4) and block data distribution (see Fig. 3). As a consequence, the framework has high compliance capability with regulatory environments in different nations, and can satisfy real business practices.

Creative Banking Infrastructure Design. This paper introduces the first framework for a Blockchain cloud banking which is universal and has many advantages compared to the conventional models. Moreover, by offering open APIs, the cloud will gather many fintech firms involved in the innovation of the financial industry. The Blockchain not only provides a secure and scalable cloud infrastructure for digital payment and banking services but also offers a feasible solution for financial inclusion expansion and coverage, especially to unbanked people via eKYC and digital banking model (see Fig. 5).

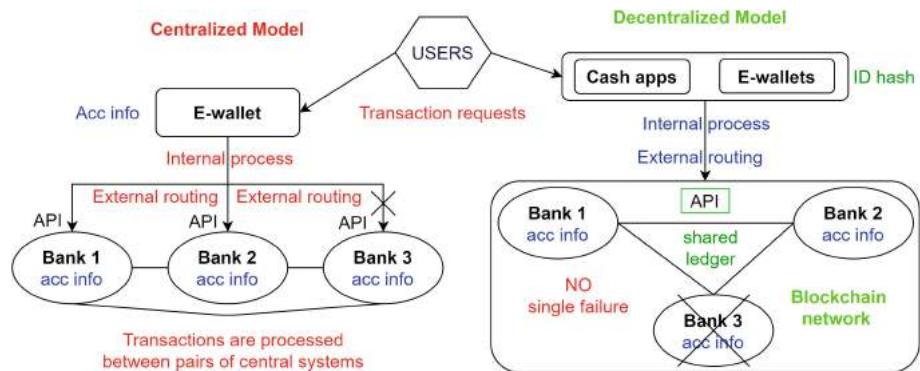


Fig. 5. Centralized vs decentralized payment applications.

7 Applications, Assessment and Conclusion

Worldwide experts have recognized that Blockchain has many application potentials in all areas of the banking and financial sector. Regarding the applications of the introduced Blockchain banking cloud, it is applicable in an wide range of financial disciplines: accounting & audit, asset tokenization, auto-invoicing, auto-governance, clearing & settlement, credit info sharing, electronic payment, micro-finance, micropayment, peer-to-peer money transfer, global money remittance, letter of credit, smartcontract-based transaction, syndicated lending. These perspectives are clearly indicated in many studies [2, 3, 6, 8, 12].

In November 11, 2020, all major media channels in crypto-currency space reported the [Ethereum split issue](#) happened at block 11234873. It observed a different chain held by a minority of the network miners whose the old Geth version (an Ethereum Blockchain client software) contained a [dormant bug](#) which was detected and reported two years ago. The issue affected deposit and withdrawal activities on Binance and many other crypto-currency exchanges. Although the bug was fixed and the whole network updated the main chain held by the majority, the event raised a big concern on centralized giant client servers. Moreover, the issue revealed that large network Blockchains possibly face the same failure due to latency and obsolescence among various groups of block keepers. Together with mining power concentration, people question decentralization and safety of

Table 1. Challenges & research opportunities in the design of blockchain protocols [13]

PREStO framework		Design of blockchain
Persistance	Weak/strong persistence	51% attack defender, large network
	Majority attacks	Blockchain-Trilemma
	Recovery mechanisms	Design of sustainability
	Governance & sustainability	Decision of governance schemes
Robustness	Fault Tolerance	Protection against collusion
	Out-of-Protocol Incentives	Well elaboration with adversaries
	Resilience to attacks	
Efficiency	Positive scale effects	Scalable design
	Throughput rate	Energy saving
	Economy of resources	Compare to conventional solutions
	Benchmarking to centralized models	
Stability	Incentive compatibility: participation, operations, applications	Incentive mechanisms
	Decentralization: entry barriers, distribution of resources	Protection against adversaries
	Fairness: reward allocation, voting-decision making	Decentralization motivation, fair distribution of resources
Optimality	Liveness	Architecture & Sybil protection
	Safety	Safety vs liveness trade-off
	Scope	Smartcontract execution
	Privacy features: public/private, permissioned/-less	

Ethereum blockchain (see safety definition on [13]). Despite limited decentralization, Tron and our Blockchain frameworks offer many advantages on efficiency, stability and optimality while being robust and persistent to various attack and collusion schemes. In particular, they provide an easy recovery possibility which is very hard on Bitcoin and Ethereum (see a completed split and rollback on Ethereum Blockchain after the [DAO attack](#)).

Let acronym our proposed Blockchain Banking Cloud as BBC. According to the systematic framework (PREStO) introduced by S. Leonardos et al. [13],

we have a brief summary (see Table 1), assessment and comparison among Ethereum, Tron and BBC in Table 2.

More rationales to the assessment Table 2, our Blockchain model utilizes reputation system, hence offers higher decentralization of network resource distribution and fairness to application developers (also value contributors on the Blockchain). In addition, by the nomination and voting mechanism, our Blockchain has sustainable governance and network expansion.

For conclusion, the paper has introduced a novel framework for Blockchain banking cloud as a fundamental infrastructure for an open API platform in banking sector, then promoting innovation in payment and financial applications. The proposed Blockchain is implementable with clearly described architecture, governance, consensus, necessary techniques and protocols. We present assessment and potential applications corresponding with published studied frameworks. Although our tentative design leads to a domestic peer-to-peer banking system, the Blockchain model can be used for international bank settlement, cross-border escrow (money remittance) network wherein a secure, sharing, reliable, synchronous ledger and transaction processing system among participants is necessary and useful.

In the further work, the author shall provide more quantitative analysis on the Blockchain banking cloud, especially on the reputation system with some experimental results from existing public Blockchain transaction data.

Table 2. Assessment among Ethereum, Tron and BBC based on PREStO.

PREStO framework	Ethereum	Tron	BBC
Persistance	High	High	
- Weak/strong	Strong	Medium	
- Recovery mechanisms	Hard	Easy	
- Governance & sustainability	Majority follow	Majority voting	
Robustness	High	High	High
Efficiency	Low	High	High
- Scalability	Extremely bad	Good	Good
- Throughput	Very low	High	High
- Energy	Consuming	Saving	
- Benchmarking	Extremely low	Medium	
Stability	Medium	Medium	High
- Incentive	Fairly	Fairly	Good
- Decentralization	Highest	Medium	High
- Fairness	Medium	Medium	High
Optimality	Medium	High	High
- Liveness	High	High	High
- Safety	Medium	High	High
- Contract execution	Slow, complex	Fast & less complex	
- Privacy	Public & permissionless	Public & semi-permissionless	

References

1. Central Banks and Distributed Ledger Technology: How are central banks exploring blockchain today? Insight report, World Economic Forum, March 2019. bit.ly/3aGt23U
2. Central Bank Digital Currencies: foundational principles and core features. Bank for International Settlements, October 2020. <https://www.bis.org/publ/othp33.htm>
3. Central Bank Digital Currency: Policy-maker toolkit. Insight report, World Economic Forum, January 2020. bit.ly/3mHDg69
4. DPoS consensus mechanism. bit.ly/3aGtb7s
5. EOS Consensus Protocol. bit.ly/37M3tws
6. Forecast: Blockchain Business Value, Worldwide, 2017–2030, Gatner’s Research. gtnr.it/3pyccZd
7. Hedera Hashgraph. <https://docs.hedera.com/guides/>
8. How Blockchain Could Disrupt Banking (2018). bit.ly/3aFq13y
9. ICO fundraising. bit.ly/2KVcWrX
10. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: OSDI 1999: Proceedings of the Third Symposium on Operating Systems Design and Implementation, pp. 173–186 (1999)
11. Castro, M., Liskov, B.: Practical byzantine fault tolerance and proactive recovery. ACM Trans. Comput. Syst. (2002). bit.ly/3aFijqk
12. Allen, S., Capkun, S., Eyal, I., et al.: Design choices for central bank digital currency: policy and technical considerations. Global Economy & Development at BROOKINGS, July 2020. is.gd/DO9Oq7
13. Leonardos, S., Reijsbergen, D., Piliouras, G.: PREStO: a systematic framework for blockchain consensus protocols. IEEE Trans. Eng. Manag. **67**(4), 1028–1044 (2020). <https://doi.org/10.1109/TEM.2020.2981286>
14. Tendermint Core. <https://github.com/tendermint/tendermint>
15. The Algorand Consensus Protocol. bit.ly/3nZGuDP
16. The Binance Chain Blockchain. <https://docs.binance.org/blockchain.html>
17. The Tangle. t.ly/sT6T
18. Do, T., Nguyen, T., Pham, H.: Delegated proof of reputation: a novel blockchain consensus. In: IECC 2019: Proceedings of the 2019 International Electronics Communication Conference, pp. 90–98, Japan (2019). bit.ly/2KQ72IM
19. Boring, P., Kaufman, M.: Blockchain: the breakthrough technology of the decade and how China is leading the way – an industry white paper. Chamber of Digital Commerce and Marc Kaufman, Partner, Rimon Law, February 2020. bit.ly/3rsgPFJ



A Robust and Efficient Micropayment Infrastructure Using Blockchain for e-Commerce

Soumaya Bel Hadj Youssef^(✉) and Noureddine Boudriga

School of Communication Engineering, University of Carthage, Tunis, Tunisia

Abstract. The rapid evolving of Internet and e-commerce technology in recent years has led to rapid growth of online shopping which has become a central part of our daily life. Micropayment systems are emerging rapidly in e-commerce sites although their lack of security and robustness. In this paper, we provide a micropayment infrastructure using the blockchain technology for securing and verifying the payment transactions achieved between buyers and sellers. Moreover, our proposed infrastructure provides ways for managing the trust level of the buyer based on his historical purchases, controlling the size of the block of tokens that will be uploaded into the blockchain network for later validation, and estimating the risk of loss to be handled. Furthermore, we propose three trust models that compute the buyer's trust value. The behavior of the buyer has an impact on the block size and the risk of loss. Our micropayment infrastructure ensures payment transaction authentication, prevention from double-spending and double-selling, prevention from tokens forging, tracing of payment transaction, protection and prevention against cyber attack, and reduction of the delay of transaction verification and the waiting time of the buyer. Finally, we conduct a simulation to evaluate the performance of our three trust models and show how the behavior of the buyer has an impact on the decision made by the auditing entity.

Keywords: Micropayment · Blockchain technology · Payment security · Trust · Risk management · e-Commerce

1 Introduction

In recent years, micropayments are increasingly important with the increase need for inexpensive and simple payment methods and the rapid development of e-Commerce in which buyers and sellers are trading with each other on the Internet. Micropayment [11] can be defined as a payment of a small amount of money which is made electronically. It can be considered as an online financial transaction that contains a very small amount of money. The most use case of Micropayments is paying for online content such as online papers, archives, music, videos, games, software downloads, tickets, stamps, mobile applications,

subscription services, e-commerce stores, and among others. Micropayment provides many advantages for the customers (such as, speed, versatility, convenience, and flexibility) and for the merchants (e.g., speed and acceptable transaction fees). In the literature, many micropayment systems have been proposed. However, these micropayment systems still present several challenges such as security and scalability.

In this context, the Blockchain technology, which is considered as one of the greatest technological advances, can be a promising solution for micropayment system due to its characteristics like distributed consensus, cryptographic transactions, anonymity, and full transparency. A blockchain can be defined as a chain of online transactions saved as a shared ledger across numerous computers on a peer-to-peer network.

Recently, the blockchain has emerged as a fundamental technology and has been applied in many areas of applications [2, 15] including data verification, integrity verification, data management, privacy and security, healthcare record storing, and education. Moreover, the blockchain has been emerged in trade finance, insurance, supply chain management, product tracing, e-commerce [6, 14], and marketing [7].

In that sense, adopting the blockchain technology for micropayment systems will bring many benefits in e-commerce thanks to the low cost of transactions and its preventive mechanisms used to prevent from some fraud and online malicious activities [13]. However, this technology does not eliminate all attacks and suffers from longer transaction confirmation time. For instance, a Bitcoin transaction requires six confirmations from miners before it is processed and the average time needed for mining a block is ten minutes. In the literature, many research works have been proposed about the combination use of the blockchain technology and micropayment. For instance, in [8], the authors introduced a cost-saving approach, which significantly reduces transaction time and storage for micropayment. Authors in [4] conducted a comparison between Stellar, Lightning Network, Raiden, and PayPal in order to evaluate the feasibility of Stellar. Furthermore, they analyzed a subset of transaction records in Stellar to enable fraud prevention in online monetary transactions. The undertaken feasibility study of Stellar indicates that the blockchain platform is viable to function as a micropayment system. In [5], the authors introduced RandpayUTXO which requires the payee's signature to be published in the blockchain. Moreover, authors implemented a Blum's 'coin flipping by telephone' problem to design a 'lottery ticket' that does not require any third party to facilitate the lottery. Authors in [9] carried out a benchmark performance analysis between Lightning Network and other-like solutions. They integrated the LN off-chain technology within an existing IoT ecosystem. Moreover, they designed a novel algorithm for payment channel fee reduction. However, the above proposed micropayment systems still lack the robustness and trustworthiness. Indeed, these works did not build a mechanism for the trust of the buyers, which can help to detect the misbehaving buyers. In addition, these works did not take interest on the evaluation of the risk of loss.

In this context, it is very required to integrate an auditing entity which computes the customer's trust level which represents a key factor in e-commerce. A lot of research works have been done about the trust management in e-commerce [10], wireless sensor networks [12], ad-hoc networks [1,3], and among others.

In this paper, we propose a robust and efficient micropayment infrastructure using the blockchain technology and an auditing entity. Our infrastructure is able to reduce the delay of payment transaction verification, manage the size of the block of tokens, and reduce the risk of loss that affects the auditing entity.

The contributions of our work are as follows. Firstly, we describe the architecture of our micropayment infrastructure by presenting the functions of the actors. Our infrastructure relies on the blockchain network and on an auditing entity which is responsible for building the blocks of tokens with a determined size. Secondly, we provide three trust models for the computation of the buyer's trust value. This will help the auditing entity to determine the future value of the buyer's trust and then estimate the risk of loss associated with the behavior of the buyer. Finally, the validation of our micropayment infrastructure is provided.

The remaining part is structured as follows. Section 2 presents the requirements that must be fulfilled to achieve a robust micropayment infrastructure. In Sect. 3, we present the actors of our micropayment infrastructure by detailing their functions and we present the different interactions between them. In Sect. 4, we present our proposed trust models for the buyer. In Sect. 5, we present the decision making of the auditing entity. In Sect. 6, the validation of our proposed micropayment infrastructure is provided by proving its robustness against attacks and the results of the simulation are presented. Finally, we conclude the paper.

2 Requirements for a Robust Micropayment Infrastructure

To ensure the robustness of our micropayment infrastructure, several requirements should be satisfied. Among the important requirements, we mention the following.

Tokens Aggregation: Aggregating tokens (i.e., small payments) into blocks before transmitting them to the blockchain network is crucial. Hence, verifying a block of tokens by the nodes of the blockchain is more efficient than verifying each token alone so that the verification delay will be reduced. Moreover, the size of the block should be well chosen in order to avoid loss.

Prevention from Double-Spending: It is very necessary that our micropayment infrastructure is able to detect double spending attacks. Each token cannot be used twice by the buyer (i.e., it cannot be spent by the buyer more than one for paying the seller). The buyer should use the token only once to pay the seller. This can be prevented by including the token identity and the buyer identity in the token.

Prevention from Double-Selling: Preventing our micropayment infrastructure from double selling attack is needed. This attack can be done by the seller. Hence, prevention from this attack can be done by including in each token the identities of the buyer and the seller and information about the product.

Prevention from Tokens Forging Attack: Preventing our micropayment infrastructure from the tokens forging attack is crucial. This attack can be achieved by generating false tokens during the transfer of tokens to the seller or during the construction of blocks or during the transmission of blocks to the blockchain network. Hence, it is needed to ensure that tokens have not been altered or manipulated by unauthorized parties. For this, it is crucial to verify and validate each token before proceeding with the payment.

Authentication of Payment Transaction: It is very important to ensure the authenticity of the transaction including the authenticity of the actors and authenticity of the tokens included in the transaction. Each receiver of a token or transaction should authenticate the origin of this token or transaction. Accordingly, including the identities of actors and signing the tokens and transactions circulated between actors is crucial.

Payment Transaction Tracing: It is very required to trace the transactions and store the histories of payment. Using the blockchain network can be an efficient solution. Moreover, the inclusion of the timestamps on every interaction achieved between the actors is very important. Accordingly, transaction tracing can be done by adding identities of the actors, information about the product, and timestamps.

Buyers' Trust Management: In order to reduce the risk of loss, it is very important to control the trust level of the buyer. This can be done by including a trusted entity in our micropayment infrastructure for managing the trust level of the buyers. In fact, the decision made by this entity depends on the behavior of the buyer.

3 Micropayment Infrastructure Based on Blockchain

In this section, we focus on the presentation of the architecture of our robust micropayment infrastructure. We first present the actors and their roles. Then, we describe the interactions between these actors, as depicted in Fig. 1.

3.1 Description of the Actors

The main actors of our infrastructure are as follows:

The Buyers: A buyer is a person who purchases the goods and services from the sellers using micropayment tokens. The buyer has an account at the financial entity. His global amount will be divided into a large number of micropayments. The buyer will receive tokens from the financial entity and will send them to the sellers.

The Sellers: A seller is the person who sells his goods and services to the buyers. He is charged of constructing a payment transaction by including the received tokens concerning a given product or service, his identity, the identity of the buyer, and information about the purchase. He is also responsible for sending the transaction to the auditing entity.

The Auditing Entity: It is a trusted third-entity whose main role consists in controlling the tokens included in the received transactions from the sellers. The tokens are grouped by buyers. The tokens corresponding to the same buyer are grouped together in blocks. He is also responsible for managing the buyer's trust and hence determining the sizes of the blocks of tokens. After forming blocks, he submits the blocks to the blockchain network.

The Financial Entity: It has three main functions consisting in: a) generating and sending the tokens to the buyers; b) paying the auditing entity after validation of the tokens; and c) providing the blockchain network with information about the generated tokens.

The Blockchain Network: It is a peer-to-peer network of computers (i.e., miners). Their main functions consist in registering, verifying, and managing the received transactions from the auditing entity.

3.2 Interactions Between Actors

This section presents the different interactions between the actors (as depicted in Fig. 1) during the tokens and transactions generation, block formation, block size determination, and token payment.

- Interaction between the financial entity and the buyer

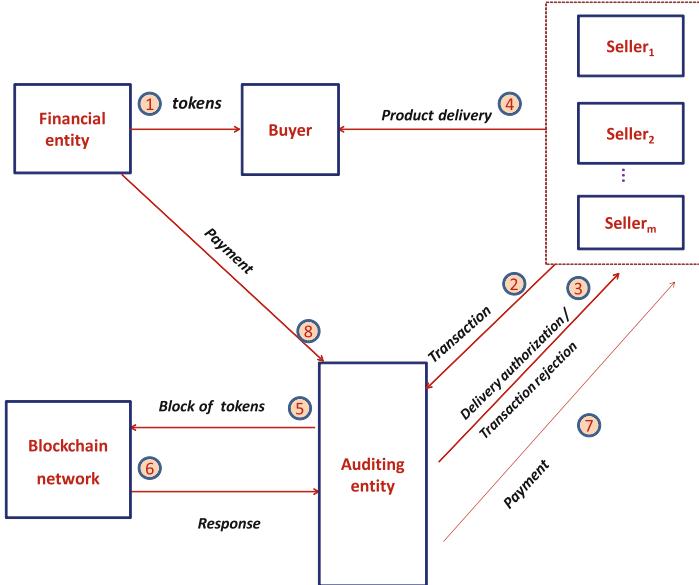
Assuming that the buyer has an account at the financial entity, the global cost is subdivided into a set of tokens having a fixed value v . When the buyer wants to buy a good or a service it sends a request to the financial entity in order to receive micropayment tokens. After request reception, the financial entity generates tokens containing the following information: a unique identifier of the token id_t , the token value v , the certificate of the financial entity $cert_{FE}$, the timestamp of the financial entity $time_{FE}$, and the signature of the financial entity. Hence, the generated token t_{FE} (message 1 as shown in Fig. 1) can be defined as follows:

$$t_{FE} = \{cert_{FE}, id_t, v, time_{FE}, sig_{FE}(id_t, v, time_{FE})\} \quad (1)$$

This token is then transmitted to the buyer.

After reception of the token from the financial entity, a verification process is performed by the buyer. The latter verifies the authenticity of the token, the financial entity's certificate, and the timestamp included in the token.

It is worth noting that the operation of verification will be performed by each receiver of a token. The authenticity of the token is ensured by verifying

**Fig. 1.** Micropayment infrastructure

the digital signature. The sender's certificate is verified by sending a request to the certificate authority. The timestamp included in the token is verified by comparing it with the timeclock of the token's receiver.

- Interaction between the buyer and the seller

When the buyer wants to buy a service from a given seller, he begins to perform a transaction with it. After verification done by the seller, the built transaction Tr contains the following information: certificate of the buyer $Cert_{buyer}$, certificate of the seller $Cert_{seller}$, a set of tokens θ needed to cover the purchase, information about the purchase inf_{purch} , the time of creation of the transaction $time_{Tr}$, and the resignature of the seller on the signature of the buyer.

$$Tr = \langle Cert_{buyer}; Cert_{seller}; time_{Tr}; \pi; sig_{seller}(sig_{buyer}(time_{Tr}; \pi; inf_{purch})) \rangle \quad (2)$$

This transaction (message 2) will be sent to the auditing entity.

- Interaction between the seller and the auditing entity

After reception of the transaction from the seller, the auditing entity firstly verifies the authentication, the certificate, and the timestamp. Secondly, it extracts the tokens included in θ and then insert them into the block of the corresponding buyer. Then, it verifies if the constructed block size is achieved or not. In fact, the number of tokens assigned to the buyer changes over time

according to the buyer's profile (i.e., trust value). Two possible cases can be presented: a) when the block reaches its size, the auditing entity sends the block to the blockchain network (message 5) for verification and still wait the result from the blockchain network; and b) if the block size is not reached, it will send a signed message to the seller ordering it to deliver the purchase (message 3). Then, the seller will deliver the purchase (message 4) to the buyer. Moreover, the auditing entity will proceed to redeem all the tokens included in the transaction (message 7).

- Interaction between the auditing entity and the blockchain network

After receiving each transaction from the seller and after formation of a block, the auditing entity uploads the built block into the blockchain network (messages 5). In fact, the auditing entity inserts the required number of tokens into the block B_{AE} according to its size S , adds its certificate $cert_{AE}$ and its timestamp $Time_{AE}$, and then signs the block with its certificate.

The formed block $Block_{AE}$ can be expressed as follows:

$$Block_{AE} = \langle cert_{AE}; Time_{AE}; B_{AE}; sig(Time_{AE}; B_{AE}) \rangle \quad (3)$$

where B_{AE} is the block containing S tokens which corresponds to the block size assigned to the buyer.

Now, after receiving the blocks from the auditing entity, the blockchain network applies the verification process to determine the validity of each token. Firstly, it verifies the authenticity, the timestamp, and the certificate of the actors. Secondly, it checks if the token is valid or not and determines the type of attack. Finally, it sends the response to the auditing entity (message 6) which consists in a block composed of the states of tokens. This result $Block_{BC}$ can be expressed as follows:

$$Block_{BC} = State(B_{AE}) = \{St_j\} \quad (4)$$

where $St_j = \langle s_j, b_j, ta_j \rangle$ ($j = 1 \dots S$) represents the state of a token included in the block of size S . In fact, each state contains the identity of the token (s_j), the binary value (b_j) (i.e., 1 for valid token or 0 for invalid token), and the type of attack (ta_j) in case of false tokens.

After receiving the result from the blockchain network, the auditing entity determines the number of invalid tokens and then computes the buyer's trust value and hence the future size of the block of tokens.

- Interaction between the auditing entity and the financial entity.

The auditing entity, after receiving the response from the blockchain network, it firstly extracts the valid tokens from the received block and then sends an invoice to the financial entity to receive payment for these valid tokens. After reception of the invoice, the financial entity will verify this invoice before proceeding to payment by sending a request to the BC network to check the validity of the identities of tokens included in the invoice. If these tokens are validated by the miners, the financial entity proceeds to pay the auditing entity (message 8).

4 Buyer's Trust Models

In this section, we are interested to present the trust models of the buyer. In fact, the auditing entity is responsible for computing the trust values of the buyers. The main roles of the auditing entity consists in: a) aggregating the tokens and forming the blocks that will be transmitted to the blockchain network; and b) determining the risk level and the future value of the tokens block size according to the computed trust value. It is very interesting to note that trust is dynamic and related to risk. For a given buyer, let S be the size of the current block of aggregated tokens to be transmitted to the blockchain network and b be the number of invalid tokens included in the previous transmitted block. In the sequel, we present our three trust models.

4.1 Neutral Trust Model

The neutral model can be represented by a linear trust function as follows:

$$T_{e^0}(b) = \frac{(S - b) + 1}{S + 2} \quad (5)$$

It is worth noting that we have $T_{e^0}(0) = \frac{S+1}{S+2}$ and $T_{e^0}(S) = \frac{1}{S+2}$.

4.2 Optimistic Trust Model

The optimistic model can be represented by the following trust function:

$$T_{e^-}(b) = 1 - \gamma_2 \times \exp(-\delta_2 \times (S - b)) \quad (6)$$

where $\gamma_2 = \frac{S+1}{S+2}$ and $\delta_2 = \frac{\log(1+S)}{S}$

We have $T_{e^-}(b) : 0 \rightarrow W$, $T_{e^-}(0) = \frac{S+1}{S+2}$ and $T_{e^-}(S) = \frac{1}{S+2}$. In addition, it has two main properties: its first order derivative is negative so that this function is decreasing and its second order derivative is negative for $b \in [0, S]$ so that $T_{e^-}(.)$ is convex.

4.3 Pessimistic Trust Model

The pessimistic model can be represented by the following trust function:

$$T_{e^+}(b) = \gamma_1 \times \exp(\delta_1 \times (S - b)) \quad (7)$$

where $\gamma_1 = \frac{1}{S+2}$ and $\delta_1 = \frac{\log(1+S)}{S}$.

We have $T_{e^+}(0) = \frac{S+1}{S+2}$ and $T_{e^+}(S) = \frac{1}{S+2}$. Moreover, it has two main properties: its first order derivative is negative so that $T_{e^+}(.)$ is decreasing and its second order derivative is positive on $[0, S]$ so that $T_{e^+}(.)$ is concave.

It is worth noting that the three functions have the same start point and end point.

Finally, one can easily show that for a given S , we have the following inequalities:

$$T_{e^+}(b) \leq T_{e^0}(b) \leq T_{e^-}(b) \quad (8)$$

5 Auditing Entity's Making Decision

In this section, we first detail how the auditing entity handles the size of the blocks of tokens it sends to the blockchain network for later computing the future trust value. Then, we present an approximation of the auditing entity's related risk to be dealt with.

5.1 Computation of the Block Size

In this subsection, we describe how our payment system is trust aware. We first detail how the auditing entity handles the size of the blocks of tokens it sends to the blockchain system using the recent buyer's trust value. The auditing entity will make his decision by updating the trust value and modifying accordingly the tokens block size S .

The decision made by the auditing entity depends on the result of the blockchain network. The new trust value is computed only when the block is invalid. For this, the auditing entity will select a trust model and will then compute the new size of the block.

In the last case, at the construction of the n th block, the auditing entity firstly determines the number of invalid tokens b_n , and secondly computes the accumulated number of invalid tokens $\Phi_n = \sum_{j=1}^n b_j$ and hence the trust function $T_e(\Phi_n)$. Then, the auditing entity computes the relative error of the trust $\delta_n = \frac{T_e(\Phi_n) - T_e(\Phi_{n-1})}{T_e(\Phi_{n-1})}$. Finally, as the relative error of the trust is equal to the relative error of the block size the auditing entity can write the following expression:

$$\frac{T_e(\Phi_n) - T_e(\Phi_{n-1})}{T_e(\Phi_{n-1})} = \frac{y_n - S_{n-1,i}}{S_{n-1,i}} \quad (9)$$

where $S_{n-1,i}$ is the size of the block at the $(n-1)$ th formation of the block and i is the buyer.

Therefore, we can deduce the value of the solution $y_n = S_{n-1,i} \times (1 + \delta_n)$.

Now, the size of the n th block when Φ_n is strictly positive is expressed as follows: $S_{n,i} = \lfloor y_n \rfloor$, where $\lfloor \cdot \rfloor$ is the floor function.

5.2 Risk Evaluation

In this subsection, we are interested to evaluate the risk of loss of money that can affect the auditing entity due to the rise of the number of invalid tokens received after the validation process performed by the miners of the blockchain network.

For instance, when the number of invalid tokens is high the risk level will be very high so that the auditing entity will be strict with these dishonest buyers. However, if the risk is negligible the auditing entity will be more tolerated. Therefore, the risk function $Risk_{n,i}$, where n is the block number and i is the buyer number, can be expressed as the difference between the amount of payment given to the seller and the amount of payment received from the financial entity after the reception of the result of the n th formed block. Hence, one can say that the risk is related to the number of invalid tokens (b_n), the percentage of compensation (λ) of the auditing entity, the initial value of the block size, and the token value (v).

It is worth noting that the formation of the n th block may be performed after m successive transactions. After this, the block will be transmitted to the blockchain network for verification. As discussed earlier, two possible results can be obtained:

- All the tokens included in the block are valid hence the auditing entity pays an amount to the seller which is equal to $S_{n-1,i}v(1 - \lambda)$ and receives an amount from the financial entity which is equal to $S_{n-1,i}v$.
- The (n) th block presents invalid tokens then the auditing entity rejects the last transaction and gives to the seller an amount equal to the sum of all the tokens included in the $m - 1$ transactions multiplied by $v(1 - \lambda)$. The auditing entity receives from the financial entity an amount equals to the sum of valid tokens included in $m - 1$ transactions multiplied by the value of the token.

6 Infrastructure Validation and Simulation

In this section, we first discuss the validation and then describe the simulation of our provided micropayment infrastructure.

6.1 Infrastructure Validation

In this subsection, we focus on proving that our infrastructure complies with the requirements discussed in Sect. 2.

Prevention from Double-Spending: Double spending can be performed by the buyers. Our payment infrastructure is prevented from this attack since each token generated by the financial entity is identified by a unique identity. In addition, the certificates of all the actors (i.e., financial entity, buyer, seller, and auditing entity) of our infrastructure are added to each token. Moreover, our infrastructure relies on the blockchain network for verifying and validating the tokens. Hence, if a token is double spent the nodes will detect this attack and will make the state of this token invalid.

Prevention from Double-Selling: The sellers are responsible for this attack by selling the product twice to the buyers. Our infrastructure is prevented from this attack. In fact, the certificates of all the actors (i.e., financial entity, buyer,

seller, and auditing entity) are included in each token. Moreover, each transaction contains information about the purchase (e.g., number of product ...).

Prevention from Tokens Forging: The sellers are responsible for the forgery attack. This attack is prevented by our infrastructure by including the certificates of the actors and the provision of the signature mechanism. In fact, each actor, after receiving a message (i.e., token or transaction), will add its certificate and will then sign its new built message by adding the necessary information. The verification of blocks, which is achieved by the nodes of blockchain network, contributes to prevent from this attack.

Payment Tracing: Traceability is guaranteed in our infrastructure through the use of the blockchain network for verifying the tokens and the auditing entity for managing the trust of buyers. Moreover, the verification of the authenticity of the token and its sender, and the timestamp included in the token is performed by each receiver of token. This will help to trace each token. In addition, the inclusion of information about the purchase will help to trace the bought products.

Buyer's Trust Management: Trustworthiness is provided by our infrastructure through the use of an auditing entity which computes the trust value of the buyer according to the result of verification of tokens received from the blockchain network. In our work, three models for the computation of the trust are provided. The selection of one of these models depends on the decision that will be made by the auditing entity which is dependent on the number of invalid tokens received from the blockchain network.

6.2 Simulation and Results

In this subsection, we are interested to evaluate the performance of our proposed trust models. For this, we assume that n buyers want to make purchases from a seller through micropayment tokens. Each buyer owns a large number of tokens received from the financial entity. In every time slot, a buyer can buy one product with a fixed frequency f . In addition, we assume that the sold products have the same cost. Moreover, we assume that a buyer can double spend the token with a double spending rate r . Furthermore, we fix the initial value of the size of the block of tokens S_0 . Finally, the three trust models are applied to show their impact on the size of the block of tokens.

The following Table 1 summarizes the used parameters.

To analyze the performance of our infrastructure, we assess the mean size of the Block of tokens \bar{S} per buyer over time by varying: a) the double spending rate; and b) the starting value of the size of the block S_0 .

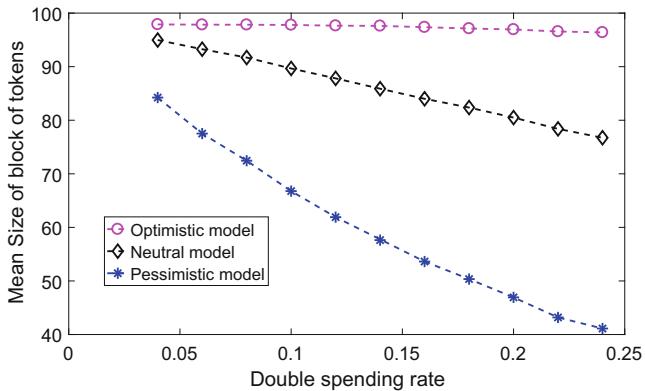
In the following, we focus on showing the results of our simulation.

As shown in Fig. 2, the mean size of the block of tokens \bar{S} is assessed with respect to the double spending rate r when one buyer and one seller are used. This is applied for the three trust models. The starting value of the size of the block of tokens S_0 is set to 100. The frequency of buying products f is equal to

Table 1. Values of parameters

Parameter	Value
Starting value of S (S_0)	100
Product price	5
Frequency of buying (f)	0.4
Number of buyers	1
Rate of double spending (r)	0.2
Auditing entity's compensating value (ε)	0.2

0.4. The price of the product is equal to 5. We observe that the mean size of the block of tokens \bar{S} decreases with the increase of the rate of double spending. This is shown for the three trust models. In addition, when we increase the rate of double spending, we show that the decrease of the optimistic model is slow while the decrease of the pessimistic model is rapid. However, the neutral model shows a moderate decrease. We note that the values of \bar{S} of the pessimistic trust model are the lowest values compared to the values obtained in the two other models. To resume, the pessimistic model is chosen when the auditing entity wants to be strict with the buyers. However, the optimistic model is applied when the buyers are honest.

**Fig. 2.** Mean size of the block of tokens w.r.t. double spending rate

As shown in Fig. 3, we assessed the mean size of the block of tokens \bar{S} in the case of the neutral trust model with respect to the cost of the product and the double spending rate r . S_0 is equal to 100 and f is set to 0.4. We notice that the mean size of the block \bar{S} decreases when the rate of double spending is increased. In addition, we note that \bar{S} decreases when we increase the price of the product. Indeed, the more we increase the rate of double spending and the

product price the less is the mean size of the block \bar{S} . We note that the highest value of \bar{S} is achieved for the lowest values of the product price and the double spending rate. Furthermore, we observe that the lowest value of \bar{S} is reached when the values of both the product cost and the double spending rate are the highest. To resume, we can say that it is encouraged to reduce the product price for the buyers having bad behavior.

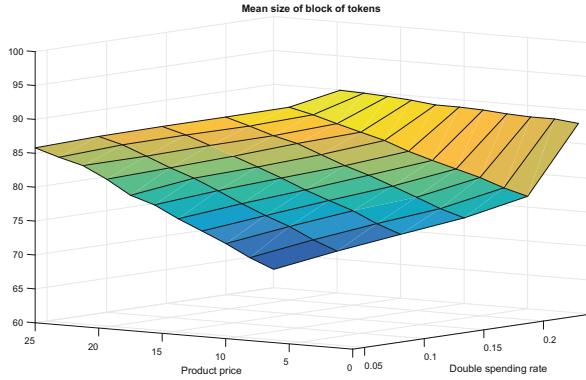


Fig. 3. Mean size of the block of tokens w.r.t. double spending rate and product cost

As depicted in Fig. 4, we assessed the mean size of the block of tokens \bar{S} with respect to the starting value S_0 . In this simulation, we are also interested to show the impact of the three trust models. The product price is set to 5 and the double spending rate is set to 0.2. It is worth noting that the more we increase the value of S_0 the more is the value of \bar{S} . Furthermore, we observe that the optimistic model shows higher values of \bar{S} than the neutral model. Moreover, the latter also shows higher values of \bar{S} than the pessimistic model. To resume, we can say that the selection of the trust model is related to the behavior of the buyer.

As depicted in Fig. 5, we evaluated the mean size of the block of tokens with respect to S_0 and the product price. In this simulation, we select the pessimistic model for evaluation. The rate of double spending is set to 0.2. We notice that the mean size of the block \bar{S} rises with the increase of the starting value S_0 and drops with the increase of the product price. Accordingly, the more is the product price and the less is the starting value S_0 the less is \bar{S} . Furthermore, the highest value of \bar{S} is reached for lowest value of product price and highest value of S_0 . To conclude, the selection of the starting value has an impact on the decision made by the auditing entity.

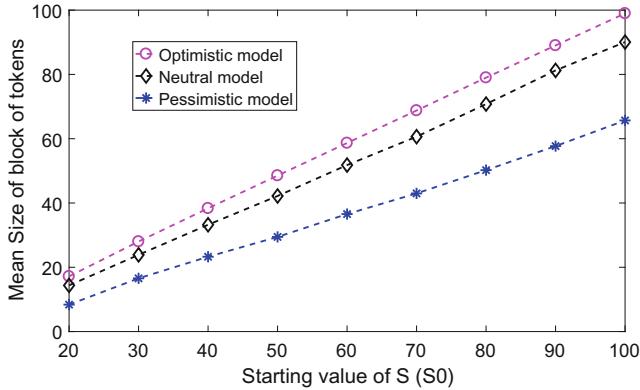


Fig. 4. Mean size of the block of tokens w.r.t. initial value of $S (S_0)$

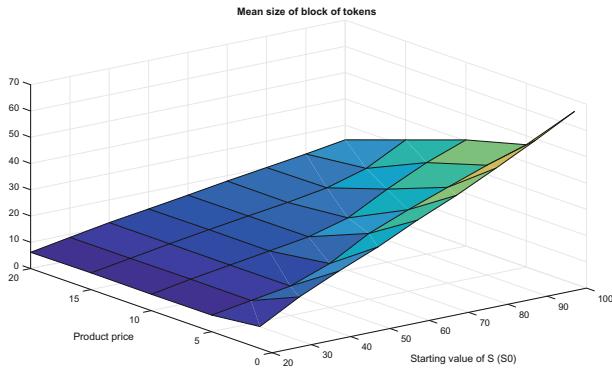


Fig. 5. Mean size of the block w.r.t. starting value of $S (S_0)$ and product cost

7 Conclusion

In this work, we presented our robust micropayment infrastructure using the blockchain technology for e-commerce. Moreover, we presented three trust models for computing the trust values of the buyer. Furthermore, we presented the decision made by the auditing entity which is based on the determination of the future value of the size of the block and the risk of loss caused by misbehavior buyers. Finally, we validated our micropayment infrastructure and analyzed the performance of our proposed trust models, by assessing the mean size of the block per buyer over time.

In the future, we plan to show the scalability of our infrastructure by considering many buyers and sellers. Moreover, we are interested to use two or more auditing entities and show their impact on the performance of our micropayment infrastructure.

References

1. Alnumay, W., Ghosh, U., Chatterjee, P.: A trust-based predictive model for mobile ad hoc network in internet of things. *Sensors* **19**(6), 1467 (2019)
2. Casino, F., Dasaklis, T.K., Patsakis, C.: A systematic literature review of blockchain-based applications: current status, classification and open issues. *Telemat. Inform.* **36**, 55–81 (2019)
3. Khan, N., Ahmad, T., State, R.: Blockchain-based micropayment systems: economic impact. In: Proceedings of the 23rd International Database Applications & Engineering Symposium (IDEAS 19), Athens, Greece (2019)
4. Khan, N., Ahmad, T., State, R.: Feasibility of Stellar as a Blockchain-Based Micropayment System. In: Smart Blockchain-2nd International Conference, Smart Block 2019, pp. 53–65. Springer, Cham (2019)
5. Konashevycha, O., Khovayko, O.: Randpay: the technology for blockchain micropayments and transactions which require recipient's consent. *Comput. Secur.* **96**, 101892 (2020)
6. Lakhani, M.J., Wang, S., UrbaÅžnski, M., Egorova, M.: Sustainable B2B E-commerce and blockchain-based supply chain finance. *Sustainability* **12**, 3968 (2020)
7. Rejeb, A., Keogh, J.G., Treiblmaier, H.: How blockchain technology can benefit marketing: six pending research areas. *Front. Blockchain* **3**(3), 1–12 (2020)
8. Rezaeibagha, F., Mu, Y.: Efficient micropayment of cryptocurrency from blockchains. *Comput. J.* **62**(4), 507–517 (2018)
9. Robert, J., Kubler, S., Ghatpande, S.: Enhanced lightning network (off-chain)-based micropayment in IoT ecosystems. *Futur. Gener. Comput. Syst.* **112**, 283–296 (2020)
10. Rofiq, A., Mula, J.M.: The effect of customers' trust on e-commerce: a survey of Indonesian customer B TO C transactions. In: Proceedings of the International Conference on Arts, Social Sciences & Technology, Penang, Malaysia (2010)
11. Verme, P.L.H.: Virtual currencies, micropayments and the payments systems: a challenge to fiat money and monetary policy? *J. Virtual Worlds Res.* **7**(3), 325–343 (2014)
12. Xiaoling, W., Huang, J., Ling, J., Shu, L.: BLTM: beta and LQI based trust model for wireless sensor networks. *IEEE Access* **7**, 43679–43690 (2016)
13. Xu, J.J.: Are blockchains immune to all malicious attacks? *Financ. Innov.* **2**(1), 1–9 (2016). <https://doi.org/10.1186/s40854-016-0046-5>
14. Yang, C.-N., Chen, Y.-C., Chen, S.-Y., Wu, S.-Y.: A reliable e-commerce business model using blockchain based product grading system. In: Proceedings of. 2019 the 4th IEEE International Conference on Big Data Analytics, Suzhou, China, pp. 341–344 (2019)
15. Zile, K., Strazdina, R.: Blockchain use cases and their feasibility. *Appl. Comput. Syst.* **23**(1), 12–20 (2018)



Committee Selection in DAG Distributed Ledgers and Applications

Bartosz Kuśmierz^{1,2(✉)}, Sebastian Müller^{1,3}, and Angelo Capossele¹

¹ IOTA Foundation, 10405 Berlin, Germany
bartosz.kusmierz@pwr.edu.pl

² Department of Theoretical Physics, Wroclaw University of Science and Technology,
Wrocław, Poland

³ Aix Marseille Université, CNRS, Centrale Marseille,
I2M - UMR 7373, 13453 Marseille, France

Abstract. In this paper, we propose several solutions to the committee selection problem among participants of a DAG distributed ledger. Our methods are based on a ledger intrinsic reputation model that serves as a selection criterion. The main difficulty arises from the fact that the DAG ledger is *a priori* not totally ordered and that the participants need to reach a consensus on participants' reputation.

Furthermore, we outline applications of the proposed protocols, including: *(i)* self-contained decentralized random number beacon; *(ii)* selection of oracles in smart contracts; *(iii)* applications in consensus protocols and sharding solutions.

We conclude with a discussion on the security and liveness of the proposed protocols by modeling reputation with a Zipf law.

Keywords: Decentralized systems · Distributed ledgers · DAG · Reputation model · Security

1 Introduction

In distributed ledger technologies (DLTs), committees play essential roles in various applications, e.g., distributed random number generators, smart contract oracles, consensus mechanisms, or scaling solutions.

The most famous example of permissionless DLT is the Bitcoin blockchain introduced in the whitepaper [23] by Satoshi Nakamoto. The blockchain enables network participants to reach consensus in a trustless peer-to-peer network using the so-called proof of work (PoW) consensus protocol. In PoW based blockchains, participants need to solve a cryptographic puzzle to issue the next block. The higher the computing power of a participant, the higher the chances to produce the next block.

In the past years, another consensus mechanism, called Proof-of-Stake (PoS), became popular. In contrast to PoW, in PoS-based cryptocurrencies, the next block's creator is chosen depending on its wealth or stake [30] and not on its computing power.

DLTs already found multiple applications in the financial sector, including online value transfer, digital assets management, data marketplace [13, 41]. Successive projects, inspired by Bitcoin, started an entirely new field of smart contracts, which are used for settling online agreements without the intermediary and the open possibility of decentralized autonomous organizations [17, 25].

Nevertheless, blockchain-based DLTs have problems, which become apparent when the network participants start using full block capacity. As the number of transactions issued by the users became significantly bigger than what can fit into the block, fees began to increase considerably. Increasing the throughput is one of the main motivations behind the DLTs' scaling problem, which leads to the (in-)famous blockchain trilemma [5]. The blockchain trilemma states that DLT can have up to two out of three desired properties: scalability, security, decentralization. The most problematic part of the trilemma for blockchains is scalability.

Proposed solutions that aim at increasing DLTs scalability and flexibility include increasing block size and issuance frequency, sidechains [7], "layer-two" solutions like Lightning Network [28], different consensus mechanisms [2], sharding [21]. A different approach is to change the underlying data structure from a chain to a more general directed acyclic graph (DAG) [29, 35].

DAGs have been adopted by a variety of DLT projects including IOTA [29], Obyte [6], SPECTRE [37], Nano [20], Aleph Zero [12]. While those projects utilize the DAG structure differently and adapted different consensus mechanisms, in most of them, transactions are graph vertexes that reference multiple previously issued transactions. This property assures that the graph is acyclic, and the data structure grows with time.

1.1 DAG Based Cryptocurrencies

An example of a project that utilizes DAGs is IOTA as described in the original whitepaper [29], which requires every transaction in the DAG, also called the *Tangle*, to reference exactly two other transactions directly. Transactions also indirectly reference other transactions, we say that y indirectly approves x if there is a directed path of references from y to x . As the ledger grows, each accepted transaction gains indirect references, which are assured by the default tip selection mechanism [18, 24, 29, 35]. The set of indirect references of a given transaction is interpreted as the number of confirming transactions and play the same role as the confirming blocks in Bitcoin, i.e., the more indirect references a transaction gains, the more likely it is to remain a part of the ledger.

The currently deployed implementation of the IOTA is a form of technology prototype [15], hereafter referred to as Coordinator-based-IOTA, and differs from the original whitepaper version. The most crucial difference is that the consensus is based on the so-called *milestones* - special transactions issued periodically by a privileged node called *Coordinator*. Every transaction referenced (directly or indirectly) by milestone is considered confirmed. While such a system is centralized, authors in [34] propose a decentralized solution dubbed *Coordicide*, referred to as post-Coordicide IOTA. One key element in Coordicide is to replace the

Coordinator with the decentralized consensus protocol called Fast Probabilistic Consensus (FPC). FPC is a scalable byzantine resistant voting scheme [3, 33] where nodes vote in rounds. In each round, participants query a random subset of online nodes in the network. Opinions in the next round depends on the received queries and a random threshold. When most of the nodes in the network use the same random threshold the system reaches unanimity. The common random thresholds enable the protocol to break meta-stable situations [32].

1.2 Contributions

Let us define a committee as a group of usually trustworthy nodes, selected to execute a special task. We assume that participants take part in a DLT that supports a reputation system. Every node can issue transactions, which from now on we call *messages*. We use the name message to indicate that those objects can include generic data and not only token transfers. The reputation system is needed as a criterion to select a subgroup of all participants and to mitigate Sybil attacks, a common threat in permissionless systems.

We concentrate on the more difficult case of DAG-based DLTs, which promise better scalability and decentralization than blockchains. Unlike blockchains, which are totally ordered by nature, DAGs lack natural “reference points” that could be used to determine the reputation of the participants. The confined structure of blockchains allows for using reputation calculated for a specific block defined on the protocol level. The complex structure of DAGs does not allow for the adaptation of an analogical rule.

This paper proposes a series of committee selection mechanisms for DAG-based DLTs with reputation system. The committee selection process runs periodically and depends on the reputation of nodes, which is stake or delegated stake. More specifically, the contributions of this paper are the following:

1. we propose several protocols to select a committee in permissionless decentralized systems;
2. we analyze the token distribution of a series of cryptocurrency projects;
3. we model the reputation distribution and analyze the security of the proposed protocols; and
4. we discuss several applications of committee selection protocols, including decentralized random number beacon, smart contracts, consensus mechanism, and sharding solutions.

1.3 Outline

The article is organized as follows. In the next section, we discuss previous works related to this paper, mainly related to different kinds of decentralized random number beacon and reputation systems. In Sect. 3, we specify the assumptions on the DLT that are required for our protocol to work. Section 4 is devoted to the committee selection, where we give three different methods of achieving consensus on the reputation values. In Sect. 5, we discuss the security of

our proposal by modeling reputation distribution with a Zipf law. Applications of our protocols to dRNG, smart contracts, consensus mechanism, and scaling are discussed in Sect. 6. Finally, Sect. 7 outlines further research direction and concludes the paper.

2 Related Work

2.1 Reputation and Committees in DLT

A reputation system in a DLT is any mapping that assigns real numbers to the network participants. Reputation can be objective when all of the participants agree on the exact values of the reputation, or subjective when different nodes have different perception of the reputation. However, for the subjective reputation to maintain its utility and play the same function as in social systems, network users should have at least an approximate consensus on its values.

All PoS consensus mechanisms induce a reputation system where the user's reputation equals staked tokens [2]. In the same way, delegated PoS (DPoS) protocols, where staked tokens are delegated to other nodes [11, 19], define a natural reputation system. Highest DPoS nodes form a committee of block validators and produce the next blocks. The consensus among the fixed-size committee is easier to achieve than in open systems. An interesting variation on DPoS systems is *mana* introduced in the post-Coordicide IOTA network [34]. This reputation system takes advantage of each issued message, which temporally grants mana to a certain node.

Multiple implementations of sharding in blockchains also require a random assignment of the block validators, e.g., [8]. If validators can not predict which shard they will be assigned, then any collusion is significantly hampered. When the network also uses a reputation system, the sharding process can be improved by assigning approximately the same reputation into each shard. An example of such protocol is RepChain [14], which assigns reputation to nodes based on their behavior in the previous rounds.

2.2 Decentralized Random Numbers Generation

Both randomness and reputation systems can improve the security, scalability, and liveness of the DLTs. A random number beacon, as introduced by Rabin [36], is a service that broadcasts a random number at regular intervals. Randomness produced by an ideal beacon cannot be predicted before being published; however, this assumption is hard to achieve in centralized systems.

The development of decentralized random number generators (dRNG) tries to address those problems. In general, a dRNG should provide unpredictable and unbiased randomness, which can not be controlled nor easily biased by a single malicious actor. There are different proposals for dRNGs in the literature.

Authors in [1] discuss the extraction of randomness from public blockchains using hashes of blocks. However, certain concerns regarding those solutions have

been raised in [26, 31]. Other proposals like RandHound and RandHerd utilize publicly verifiable secret sharing schemes [39] to generate a collective key shared between committee nodes. If more than a certain threshold of partial secrets are published, the network can recover the random number, i.e., (t, n) -threshold security model. Other proposals include smart contracts, e.g., RANDAO used in Ethereum [10]. Security in such solutions is achieved with the risk of fund confiscation.

An interesting research direction that can improve security and prevent manipulation of the generated randomness are verifiable delay functions (VDFs) [1]. VDFs take a fixed amount of time to compute, can not be parallelized but can be verified quickly [27, 40]. However, VDF calculations are costly. Moreover, this approach requires further research to ensure honest users have access to the fastest application-specific integrated circuits (ASICs) specialized in a given VDF.

3 Protocol Overview and Setup

We propose a protocol for committee selection in DAG-based DLTs with a subjective reputation system. Every node has a reputation, but a priori, there is no perfect consensus on the values of the reputation. We assume that each vertex of a DAG determines the view of the reputation, and subjectivity comes from the fact that there is no unique way of choosing the “reference vertex”. For example, if nodes adopt the simple rule *reputation should be calculated based on the most recent received message*, then due to network delay, two different nodes disagree on which is the most recent.

The committee selection is the process of appointing nodes with a sufficiently high reputation.

To perform the three phases above, we require the underlying DAG to verify the following properties.

- P1 The DAG grows in time, and incoming messages are new vertexes of the DAG.
- P2 The DAG allows for the message exchange of *application messages* (optional).
- P3 The DAG is immutable and provides a strict criterion for an approximate time of message creation.
- P4 The subjective reputation of the nodes can be read from DAG and is determined by the vertex (different nodes can read reputation from different vertexes).
- P5 Messages in the DAG are signed by the nodes and can not be counterfeited (i.e., a malicious node can not fake origin of the message).

Immutability, P3, guarantees that no message can be removed from the ledger nor new messages can be added in a part of the ledger that suggests it was issued in the past (by attaching it deep into the DAG). Property P3 can be achieved using any of the following methods:

- (i) Messages are equipped with *enforceable* timestamps. Messages with wrong timestamps are rejected by the network.
- (ii) Special *partial order generating* messages are periodically issued into the DAG.

Enforceable timestamps mentioned in (i) can be achieved when honest nodes automatically reject messages with timestamps too far in the past or the future. A certain level of desynchronization must be allowed to account for the network delay and differences in local clocks. However, we require that the network has a specific bound above which no message with lower timestamp will be accepted into the ledger (even accounting for network delays). In the edge cases of timestamps, when part of the network thinks that timestamp is valid and other does not, it is necessary to run a certain kind of consensus mechanism.

Partial ordering generating (POG) messages, (ii), can be a result of consensus among the nodes, e.g., nodes vote on them. Other options are *proof-of-authority* type of consensus where privileged “validator” nodes issue those POG messages. An example of this consensus type is the Coordinator-based IOTA [15], where a particular entity called *Coordinator* issues milestones. Similarly, Obyte uses main chain transactions (MCT), which are indicated by trusted *witnesses* [6]. Note that when a node issues a message after the i th milestone/MCT, it can be approved only by a $(i+1)$ th order generating message. This procedure generates a certain kind of logical timestamps provided by the milestones/MCT.

There are two natural ways of determining the reputation value:

- (i) the reputation is summed over all of the messages approved by a given message;
- (ii) if timestamps are available, the reputation can be summed over all of the messages with timestamps smaller than DAG vertex’s timestamp.

An example of the DAG’s reputation calculated using the method in (i) is presented in Fig. 1.

4 Committee Selection

When all users have the same view on the reputation, a natural choice for the committee is to take the top n reputation nodes.

An alternative option is to perform a lottery with a probability dependent on the reputation of nodes. The list of lottery participants consists of $k > n$ nodes with the highest reputation. Using the last random number X produced by the previous committee, nodes calculate the coefficients:

$$q_i = f(X, id_i) \cdot \frac{r_i}{\sum_{j=1}^k r_j},$$

where f is a cryptographic hash function with values in $[0, 1]$, i is the index of a node, r_i its reputation, and id_i is its identifier. Then, the committee members would be the n nodes with the highest q_i .

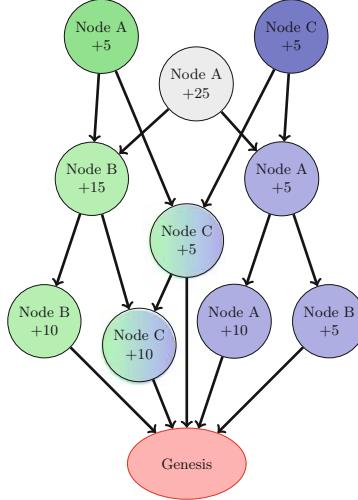


Fig. 1. Each message grants a particular reputation to one node. The reputations are summed over all of the messages approved by a given message. The reputation calculated for the dark blue message takes into account all blue messages: Node A: 15, Node B: 5, Node C: 20. The reputation for the dark green message is obtained by summing over all green messages: Node A: 5, Node B: 25, Node C 15.

Cryptocurrency projects are known for their high concentration of hashing power or (staked) tokens, and opening the possibility of low reputation nodes being members of the committee may lead to a decrease in security. Thus, we recommend selecting the top n reputation nodes. We discuss this issue in more detail in Sect. 5.

In the following, we present three different methods to find consensus on the nodes' reputations and, therefore, on the members of the committee. We describe the methods for DAGs with timestamps; the corresponding versions for DAG with POG messages are straightforward adaptations. We assume that we want to select the committee at time t_C , and D is the bound for accepting dRNG messages in the DAG, i.e., no honest node will accept dRNG messages with timestamps different by more than D from its local time.

4.1 Application Message

The first method of committee selection requires all nodes interested in the committee participation to prepare a special *application* message. This message determines the value of the reputation of a given node by its timestamp. Application messages can be submitted within an appropriate time window.

Let us denote the application time window length by Δ_A , then the time window is

$$[t_C - D - \Delta_A, t_C - D].$$

Application messages are used to extract only the reputation of the issuing node, i.e., the reputations of two potential committee members are deduced from different messages. If a node sends more than one application message, the other participants do not consider this node for the selection process. Since messages require node's signatures, this type of malicious behavior can not be imitated by an attacker.

This method of committee selection has the advantage that only online nodes can apply. Moreover, if a particular node does not want to participate in the committee, e.g., due to planned maintenance, insufficient resources, it can decide not to apply. In general, applications should not be mandatory as it is hard to enforce.

The committee selection procedure is open, and any node can issue an application message. However, the committee is formed from top reputation nodes, and low reputation nodes are unlikely to get a seat. Applications issued by low reputation nodes are likely to be not only redundant but possibly problematic when the network is close to congestion. Thus we propose the following improvements to the application process.

A node is said to be M_ϕ if, according to its view of reputations, it is among the top ℓ reputation nodes. Then nodes produce application messages according to the following:

If a node x is M_{2n} then it issues an application at the time $t_C - D - \Delta_A$.

For $k > 2$ a node x which is $M_{n \cdot k}$ but not $M_{n \cdot (k-1)}$ submits a committee application only if at the time $t_C - D - \Delta_A / (\ell - 1)$ (according to x 's local time perception) there is less than n valid application messages with stated reputation greater than the reputation of x . The timestamp of such application message should be $t_C - D - \Delta_A / (\ell - 1)$. An example of pseudocode for scheduling the send of an application message is presented in Algorithm 1.

4.2 Checkpoint Selection

We introduce checkpoints, similar to POG messages mentioned in (ii). The checkpoint is unpredictably obtained from ordinary messages, i.e., it is impossible to say in advance that any particular message will become a checkpoint at the time of issuance. To achieve that, we require a single random number X .

The reputation of the nodes is calculated for this checkpoint message. Moreover, checkpoints can suggest which nodes are online. For example, using the following rule: if the last message in a past cone of checkpoint issued by a node x is older than a certain threshold, we consider this node offline.

The random number X can be the last random number produced by the previous committee, e.g., using the dRNG described in Subsect. 6.1, or can come from a different source. If the random number X is revealed at time t_C , the time the committee selection should start, the checkpoint is the first message with a timestamp smaller than

$$t_C - D - X.$$

Possible ties are broken with lower hash.

Algorithm 1: Application message send scheduler.

```

require : nodes_reputation(time) → descending ordered list of nodes'
          reputation at input time;
          get_position(ID, ordered_list_of_nodes) → ID's position on the
          given list;
          get_reputation(ID, time) → ID's reputation at the given time;
          application_msg_with_rep_higher_than(reputation) → amount
          of application messages in DAG with reputation higher than
          input;

input :  $t_C$ : time of committee selection;
          $D$ : bound for accepting dRNG messages in the DAG;
          $\Delta_A$ : application time window length;
         max_k: maximum index value for sending an application msg;
         n: size of the committee;
         my_ID: ID of the node;

1 if ( $time = t_C - D - \Delta_A$ ) then
2    $k \leftarrow 2$ 
3   while  $k \leq max\_k$  do
4     wait_until( $t_C - D - \Delta_A/(k - 1)$ )
5     rep_list ← nodes_reputation( $t_C - D - \Delta_A/(k - 1)$ )
6     my_ell ← get_position(my_ID, rep_list)
7     my_rep ← get_reputation(my_ID,  $t_C - D - \Delta_A/(k - 1)$ )
8     if ( $application\_msg\_with\_rep\_higher\_than(my\_rep) \geq n$ ) then
9       return
10    if ( $my\_ell \leq k * n$ ) then
11      send_application_message()
12      return
13     $k \leftarrow k + 1$ 
14 return

```

The role of the random number X is to improve security. If no participant, including a potential attacker, can predict the timestamp of the future checkpoint, successful attacks are hindered or made impossible. In the following, we describe one attack scenario in the absence of the random factor X .

The reputation changes, and it is possible that an attacker at one point in time had a high reputation but lost it in the meantime. Then, an attacker tries to place checkpoint in the favorable (for him/her) part of the DAG. Without the random factor X , the timestamp of checkpoint would be predictable $t_C - \Delta$. An attacker could mine message with this timestamp and minimal hash. By placing it strategically in the DAG, an attacker could manipulate reputation obtained from summed over all of the messages approved by a checkpoint (points P3 and (i)). This type of manipulation is presented in Fig. 2.

Note that when the DAG is equipped with POG messages, then one of the POG messages can be used as a checkpoint.

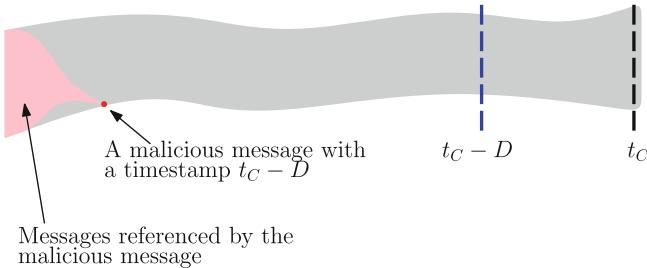


Fig. 2. An example of a checkpoint manipulation in the absence of the random factor X . A malicious transaction (red) with a timestamp $t_C - D$ is issued “deep” in the DAG, even though honest transactions with these timestamps are placed near the blue dashed line. This placement of checkpoints could promote attacker’s and diminish other user’s reputation as only contributions from the Pink “cone” would be used.

Algorithm 2: Committee Selection Scheduler for Checkpoint Selection Method.

```

require : nodes_reputation(time) → descending ordered list of nodes’
reputation at input time;
get_random_number() → global random number;
top_nodes(n, rep_list) → top n nodes in terms of reputation;
input :  $t_C$ : time of committee selection;
D: network delay;
n: size of the committee;
output : committee: list of nodes selected as committee members;
1 committee ← {}
2 if time =  $t_C$  then
3    $x \leftarrow \text{get\_random\_number}()$ 
4   rep_list ← nodes_reputation( $t_C - D - x$ )
5   committee ← top_nodes(n, rep_list)
6 return committee

```

Algorithm 2 describes the pseudocode for determining the committee with the checkpoint selection method.

4.3 Maximal Reputation

Maximal reputation from an interval approach is a combination of application message and checkpoint selection. The reputation value of the node x is the maximal reputation calculated from all of the messages in the interval $[t_C - D - \Delta_A, t_C - D]$. This approach does not require the node x to issue any special message. Note that for $\Delta_A = 0$, it reduces to checkpoint selection without random factor.

However, the method has higher computational complexity since it requires the computation of reputation for all messages issued in $[t_C - D - \Delta_A, t_C - D]$.

5 Threat Model and Security

The fact that the committee is only a subset of all the nodes may decrease the robustness against malicious actors. The security of the protocol depends on the size of the committee but also on the way the reputation is distributed among the nodes.

5.1 Zipf Law

Different protocols might define reputation and methods of gaining it differently. This affects the concentration of reputation and makes it impossible to perform an analysis in full generality. For this reason we propose to model the distribution of reputation using Zipf laws.

Zipf laws satisfy a universality phenomenon; they appear in numerous different fields of applications and have, in particular, also been utilized to model wealth in economic models [16]. In this work we use a Zipf law to model the proportional reputation of N nodes: the n th largest value $y(n)$ satisfies

$$y(n) = C(s, N)^{-1} n^{-s}, \quad (1)$$

where $C(s, N) = \sum_{n=1}^N n^{-s}$, N is the number of nodes, and s is the Zipf parameter. A convenient way to observe a Zipf law is by plotting the data on a log-log graph, with the axes being $\log(\text{rank order})$ and $\log(\text{value})$. The data conforms to a Zipf law to the extent that the plot is linear, and the value of s can be found using linear regression.

In the Fig. 3 we present the distribution of the richest accounts for a series of cryptocurrency projects for the top holders, which might be considered for the committee. We observe that most of them resemble Zipf laws. Table 1 contains estimations of the corresponding coefficients of Zipf law. Note that for PoS-based protocols the reputation of a node can, to some extent, be approximated by the distribution of the tokens. Post-Coordicade IOTA uses a reputation system called *mana* that shares some similarities with a delegated PoS, see [34], and it is reasonable to assume that the future distribution of mana would be Zipf like.

5.2 Overtaking the Committee

We assume that an adversary possesses $q\%$ of the total reputation and that it can freely distribute this reputation among arbitrary many different nodes. Honest nodes share $(1 - q)\%$ of the reputation. We assume that the reputation among the honest nodes is distributed according to a Zipf law.

Overtaking of the committee occurs when the attacker gets t threshold committee seats; the exact value of t might depend on the particular application of the committee.

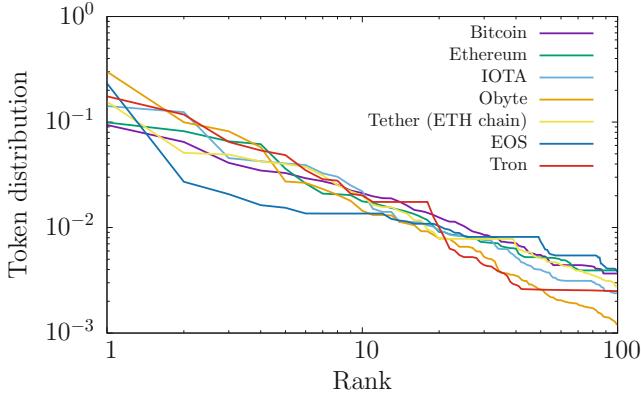


Fig. 3. The token distribution of selected cryptocurrency projects on a log-log scale (June 2020).

Table 1. Coefficients of zipf distribution with the best fit to the token distribution of given cryptocurrency project. method: linear regression on a log-log scale (June 2020).

Name	Zipf coefficient
Bitcoin	0.7628
Ethereum	0.756786
IOTA	0.934275
Obyte	1.14361
Tether (ETH chain)	0.815054
EOS	0.536744
Tron	1.02043

The cheapest way for an attacker to obtain t seats in the committee is to create t nodes with $y(n-t+1)(1-q)$ reputation each. The critical q_c that allows to get t seats in the committee is, therefore, given by

$$q_c = t \cdot y(n-t+1)(1-q_c). \quad (2)$$

Equivalently,

$$q_c = \frac{t \cdot y(n-t+1)}{1 + t \cdot y(n-t+1)}. \quad (3)$$

In Fig. 4 we present the critical value q_c for reaching the majority in the committee.

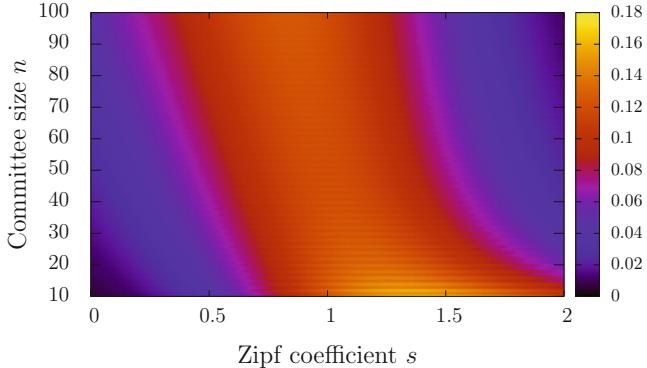


Fig. 4. Minimal amount of reputation required to overtake the committee for different committee sizes and token distribution modeled with a zipf distribution, threshold $t = \lfloor n/2 \rfloor + 1$, and number of total nodes $N = 1000$.

6 Applications

In this section, we describe a few applications of the proposed committee selection protocols.

6.1 Decentralised Random Number Generator

Committee selection protocols allow for the construction of a fully decentralized random number beacon embedded in the DAG ledger structure. An advantage of such an approach is that it relies only on the distributed ledger and does not require any interaction outside the ledger.

Publication of the random numbers would naturally take place in the ledger, where possible interruptions are publicly visible. Moreover, propagation of the random numbers would use the same infrastructure as the ledger.

Gossiping among nodes in the network would decrease the committee nodes communication overhead as they would not need to send randomness to each interested user. Similarly, if the proposed protocol requires a distributed key generation (DKG) phase (or any other setup phase), messages should be exchanged publicly in the ledger. Such an approach allows for the detection of malicious or malfunctioning committee members who did not participate in the DKG phase. After time D , a lack of corresponding messages can be verifiable proven using the ledger.

This procedure increases robustness in the case of DKG phase failure. If such failure is detected, the committee selection is repeated until the DKG phase is successful. Any iteration of the selection process may not take into account the nodes that did not participate in the DKG phase previously. Moreover, if all communication is verifiable on the ledger, such nodes can be punished, e.g., by a loss of their reputation.

Note that if the dRNG protocol uses a (t, n) -threshold scheme, i.e., n committee nodes publish their parts of the secret in the form of a beacon message and if t or more beacon messages are published then the next random number can be revealed. The procedure of obtaining the random number from the beacon messages requires specific calculations, e.g., Lagrange interpolation. To save every node from performing those calculations, special nodes in the network can gather beacon messages and publish *collective beacon* messages which already contain the random number. The public can then verify these collective beacon messages against the collective public key.

6.2 Smart Contract Oracles

One main limitation of smart contracts is that they cannot access data outside of the ledger. So-called oracles try to address this problem by providing external data to smart contracts. When multiple parties engage in a smart contract, they have to determine an oracle. An obvious solution is for contracting parties to agree upon the oracle. However, this poses certain problems as oracles may be centralized points of failure and because the different parties have to find consensus on the choice of the oracle for each contract. Moreover, contractees must know in advance that the selected oracles are going to provide reliable services upon contract expiry.

The committee selections proposed in Sect. 3 can be used to improve the security and liveness of smart contracts. For instance, if oracles gain reputations that are recorded in the ledger, then oracles can be determined using the proposed methods. If the committee selection process occurs only upon the contract termination, then the majority, if not all of the selected oracles still provide services. A similar approach is adopted in blockchain-based Chainlink [9]. Our paper allows for using analogical methods in DAG-based DLTs.

This protocol would also be user friendly as contractees are not required to know classifications of oracles. Note that specialization is very likely to occur, and different oracles probably will deliver different types of data depending on the industry and requirements. For example, sports bets are settled with *sports reputation*, whereas events in a stock market may use *financial reputation*.

Further nodes can modify smart contracts themselves to modify the weight of oracles vote power or even make outcomes depend on the oracles' opinion distribution.

6.3 Consensus and Sharding

The problem of consensus is much simpler in closed systems, where the number of participants is known and does not change. Unanimity algorithms that work in such an environment include [4, 22, 38]. However, the closed nature of such protocols makes them not very relevant for decentralization. PoW is open for any user who is willing to solve the cryptographic puzzle; the network does not require any prior knowledge of the user. An intermediary step between entirely open and permissionless networks is a system where each user is allowed to set

up a node and collect reputation, but only the most reliable nodes contribute to the consensus protocol. An example of such a protocol is EOS. In EOS delegated stake plays the role of the reputation. EOS blockchain database expands as a committee of 21 validators with the highest DPoS produce blocks. Validators use a type of asynchronous Byzantine Fault Tolerance to reach consensus among themselves and propagate the new blocks to the rest of the network [19].

An illustration of the procedure of establishing consensus based on the fixed-size closed committee in open and permissionless systems is in Fig. 5.

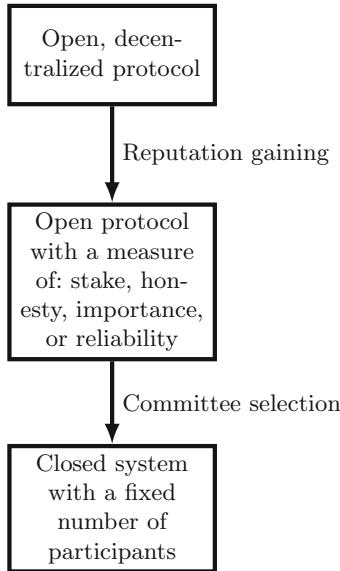


Fig. 5. Process of finding fixed-size closed committee in open permissionless systems with reputation.

Most of the DLTs with committee based consensus use either pre-selected committees or a blockchain as the underlying database structure (EOS [19], TRON [11]). Our committee selection protocols allow for using similar methods in a more general DAG structure (as long as DAG satisfies conditions P1–P5). On the same note, proposed methods can also improve scaling through sharding solutions. It is straightforward to assign validators to each shard based on their reputation. Similarly, as in the case of RepuChain [14], our solutions can assure that each shard has approximately the same total validator reputation.

7 Conclusion and Future Research

In this article, we proposed a committee selection protocol embedded in a DAG distributed ledger structure. We require the DAG to be equipped with an identity and reputation system. We further assumed that the ledger is immutable after some time, i.e., no transaction can be subtracted from the ledger; no transactions suggesting that it was issued a long time ago can be added to the ledger. These assumptions can be achieved by enforcement of approximately correct timestamps or POG transactions.

We further discussed methods of reading the reputation from the DAG. Methods include: (1) reputation summed over all of the messages approved by a given vertex in the DAG; (2) reputation summed over all of the messages with timestamps smaller than the timestamp of a given message.

Based on that we proposed and discussed the following methods of committee selection: application message, checkpoint selection, and maximal reputation method.

Furthermore, we analyzed the token distribution of a series of cryptocurrency projects, which turned out to follow Zipf distribution with a parameter $s \leq 1$. We used this fact to model reputation and analyzed the security of our proposal. Then, we focused on the applications of our committee selection protocols. We proposed an application to produce a fully decentralized random number beacon, which does not require any interaction outside of the ledger. Then, we showed how it could improve the oracle in smart contracts, and finally, we discussed possible advancements in consensus mechanism and sharding solutions.

Interesting research directions to further improve security and liveness of the proposed protocols might involve the use of backup committees. These solutions could be used when the primary committee is not fulfilling its duties. For instance, an obvious, although more centralized option, is a pre-selected committee controlled by the community or consortium of businesses interested in a reliable protocol. A different option is to select another reputation based committee from nodes with a reputation index $\{n+1, \dots, 2n\}$. However, the security of this solution is debatable, as it might be easier for an attacker to overtake it. Figure 4 shows that for $n = 20$ and $t = 11$ an attacker can overtake the committee with as little as 10% of the reputation. The value of the secondary committee would be even lower.

Other improvements to the security of the committee include giving multiple seats to the top reputation nodes in the committee, e.g., nodes from $\{1, \dots, n/2\}$ would get double or triple identities in the committee. These adaptations increase the reputation requirement to overtake the committee. Other improvements of the liveness can include recovery mechanisms when the committee fails to deliver.

We hope that all mentioned improvements to the security and liveness will stimulate further research on this topic.

References

1. Bonneau, J., Clark, J., Goldfeder, S.: On bitcoin as a public randomness source. Cryptology ePrint Archive, Report 2015/1015 (2015). <https://eprint.iacr.org/2015/1015>
2. Buterin, V., Griffith, V.: Casper the Friendly Finality Gadget. ArXiv e-prints, page arXiv:1710.09437, October 2017
3. Capossele, A., Mueller, S., Penzkofer, A.: Robustness and efficiency of voting consensus protocols within byzantine infrastructures. Blockchain: Res. Appl. **2**(1), 100007 (2021). <https://doi.org/10.1016/j.bcre.2021.100007>. <https://www.sciencedirect.com/science/article/pii/S2096720921000026>. ISSN 2096–7209
4. Castro, M., Liskov, B.: Practical byzantine fault tolerance. In: Proceedings of the Third Symposium on Operating Systems Design and Implementation, OSDI 1999, pp. 173–186. USENIX Association, USA (1999)
5. Chu, S., Wang, S.: The curses of blockchain decentralization (2018)
6. Churymov, A.: Byteball: A decentralized system for storage and transfer of value (2016)
7. Croman, K., et al.: On scaling decentralized blockchains. In: Clark, J., Meiklejohn, S., Ryan, P., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016, pp. 106–125. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53357-4_8
8. Dang, H., Dinh, T.T.A., Loghin, D., Chang, E.-C., Lin, Q., Ooi, B.C.: Towards scaling blockchain systems via sharding. In: Proceedings of the 2019 International Conference on Management of Data, SIGMOD 2019, pp. 123–140. Association for Computing Machinery, New York (2019)
9. Ellis, S., Juels, A., Nazarov, S.: Chainlink a decentralized oracle network (2017)
10. Ethereum Foundation. RANDAO: A DAO working as RNG of Ethereum
11. TRON Foundation. Tron advanced decentralized blockchain platform (2017). https://tron.network/static/doc/white_paper_v_2.0.pdf
12. Gagol, A., Leśniak, D., Straszak, D., Świetek, M.: Aleph: Efficient Atomic Broadcast in Asynchronous Networks with Byzantine Nodes. arXiv e-prints, page arXiv:1908.05156, August 2019
13. Giudici, G., Milne, A., Vinogradov, D.: Cryptocurrencies: market analysis and perspectives. J. Ind. Bus. Econ. **47**(18), 1972–4977 (2020)
14. Huang, C., et al.: Repchain: A reputation based secure, fast and high incentive blockchain system via sharding. CoRR, abs/1901.05741 (2019)
15. IOTA Foundation. IOTA Reference Implementation. Github
16. Jones, C.I.: Pareto and Piketty: the macroeconomics of top income and wealth inequality. J. Econ. Perspect. **29**(1), 29–46 (2015)
17. Kim, H., Laskowski, M., Nan, N.: A first step in the co-evolution of blockchain and ontologies: towards engineering an ontology of governance at the blockchain protocol level. SSRN Electron. J. (2018)
18. Kuśmierz, B., Sanders, W.R., Penzkofer, A., Capossele, A.T., Gal, A.: Properties of the tangle for uniform random and random walk tip selection. In: 2019 IEEE International Conference on Blockchain (Blockchain), pp. 228–236 (2019)
19. Larimer, D.: EOS.IO White Paper (2017). <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>
20. Colin, L.: Nano: A feeless distributed cryptocurrency network (2017)
21. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 17–30. ACM (2016)

22. Miller, A., Xia, Y., Croman, K., Shi, E., Song, D.: The honey badger of BFT protocols. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS 2016, pp. 31–42. Association for Computing Machinery, New York (2016)
23. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system (2008)
24. Penzkofer, A., Kusmierz, B., Capossele, A., Sanders, W., Saa, O.: Parasite chain detection in the IOTA protocol. In: Anceaume, E., Bisiére, C., Bouvard, M., Bramas, Q., Casamatta, C. (eds.) 2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020), vol. 82, pp. 8:1–8:18 (2021). <https://doi.org/10.4230/OASIcs.Tokenomics.2020.8>. <https://drops.dagstuhl.de/opus/volltexte/2021/13530>. ISSN 2190–6807
25. Peters, G.W., Panayi, E.: Understanding modern banking ledgers through blockchain technologies: future of transaction processing and smart contracts on the internet of money. In: Tasca, P., Aste, T., Pelizzon, L., Perony, N. (eds.) Banking Beyond Banks and Money. New Economic Windows, pp. 239–278. Springer, Cham (2016)
26. Pierrot, C., Wesolowski, B.: Malleability of the blockchain’s entropy. Cryptogr. Commun. **10**, 211–233 (2017)
27. Pietrzak, K.: Simple verifiable delay functions (2018). <https://eprint.iacr.org/2018/627>
28. Poon, J., Dryja, T.: The bitcoin lightning network: Scalable off-chain instant payments (2016)
29. Serguei Popov. The tangle (2015)
30. Popov, S.: A probabilistic analysis of the Nxt forging algorithm. Ledger **1**, 69–83 (2016)
31. Popov, S.: On a decentralized trustless pseudo-random number generation algorithm. J. Math. Cryptol. 37–43 (2017)
32. Popov, S.: Coins, walks and FPC. Youtube (2020)
33. Popov, S., Buchanan, W.J.: FPC-BI: Fast Probabilistic Consensus within Byzantine Infrastructures (2019). <https://arxiv.org/abs/1905.10895>
34. Popov, S., et al.: The Coordicide (2020). https://files.iota.org/papers/20200120_Coordicide_WP.pdf
35. Popov, S., Saa, O., Finardi, P.: Equilibria in the Tangle. ArXiv e-prints. [arXiv:1712.05385](https://arxiv.org/abs/1712.05385). December 2017
36. Rabin, M.O.: Transaction protection by beacons. J. Comput. Syst. Sci. **27**(2), 256–267 (1983)
37. Sompolinsky, Y., Lewenberg, Y., Zohar, A.: Spectre: a fast and scalable cryptocurrency protocol. Cryptology ePrint Archive, Report 2016/1159 (2016)
38. Stathakopoulou, C., David, T., Vukolic, M.: Mir-BFT: High-Throughput BFT for Blockchains, June 2019
39. Syta, E., et al.: Scalable bias-resistant distributed randomness. In: 2017 IEEE Symposium on Security and Privacy (SP), pp. 444–460 (2017)
40. Wesolowski, B.: Efficient verifiable delay functions (2018). <https://eprint.iacr.org/2018/623>
41. Zheng, Z., Xie, S., Dai, H.-N., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **1**, 1–25 (2016)



An Exploration of Blockchain in Social Networking Applications

Rituparna Bhattacharya, Martin White^(✉), and Natalia Beloff

University of Sussex, Falmer, Brighton, UK

{rb308,m.white,n.beloff}@sussex.ac.uk

Abstract. Human beings have an innate tendency to socialize and the Internet gives us the opportunity to overcome geographical barriers and communicate with people located even at the opposite side of the world. Computers that started the journey of digital communication with simple emails and bulletin board systems in the eighties of the previous century have manifested their potential of connecting people across the globe with the evolution of social networking. Social networking has emerged as effective tools for bringing people of similar disposition or needs, whether professional or personal, under the same platform enabling interaction and sharing between peers. However, these services operate in a centralized environment. Blockchain, on the other hand, is revolutionizing the process of sharing things digitally between peers in a decentralized and secured fashion. In this paper, we investigate how blockchain can be applied to social networking services and review some of the currently existing social networking applications that utilize blockchain. We then propose our social networking application, e-Mudra, built on the principles of blockchain technology that facilitates peer-to-peer exchange of leftover foreign currency from international travel.

Keywords: Blockchain · Leftover foreign currency exchange · Peer to peer communication · Social networking

1 Purpose

Social Networking can be envisaged as having reshaped the way people interact with each other and exchange content and information through the Internet with peers dispersed across the globe. Facebook, Twitter, Instagram are names popular amongst users situated in the remotest corner. Technologists have already started exploring the possibilities of utilizing blockchain in the avenues of social networking. Obsidian Messenger, Nexus, Indorse, Synereo, Steemit, Akasha applications are some of the instances of the endeavors made in this context. In this paper, we study the impact of blockchain in social networking applications, in general, and how the notions of blockchain and social networking can be applied together to solve the problem of exchanging leftover foreign currencies from an international trip profitably and conveniently, in particular, by presenting our e-Mudra platform.

This paper is organized as follows: In Sect. 2, we give an overview of blockchain technology; we then discuss about the social networking applications and their challenges; next we include a description of how blockchain can benefit social networking applications and a review of existing blockchain-based social networking applications; we also introduce the challenge of exchanging cash leftover foreign currency from international trips that can be alleviated using blockchain and social networking; in Sect. 3, we outline our proposed e-Mudra application; in Sect. 4, we include the results of the study and finally we conclude in Sect. 5.

2 Background

In this section, we briefly discuss about blockchain technology and then focus on social networking sites and shared economy. We then explicate the impact of blockchain on social networking, blockchain-based social networking applications currently available and the problem associated with cash leftover foreign currency exchange after any international tour.

2.1 A Brief About Blockchain

One of the cynosures of current disruptive technologies in the computing world that has invoked much attention from both academia and industry is blockchain. Devised by Satoshi Nakamoto more than a decade ago, blockchain is the fundamental pillar on which Bitcoin, the forerunner of an era of cryptocurrencies, has been built upon [1].

Prior to the invention of blockchain, peer to peer transactions of digital currencies following a decentralized pattern suffered from the issues of Double Spending and the Byzantine General's problem.

Unlike cash payments between peers, digital transactions can be forged easily and trust has to be enforced by an intermediary, that is, a centralized system such as a bank, but this requires service fees charged by those institutions. In a decentralized digital network with thousands of members dispersed worldwide where members are unknown to each other, it is difficult to reach an agreement ensuring the validity of transactions shared by them. Blockchain resolves both these issues. It maintains a transparent, immutable, shared distributed ledger which is a sequentially threaded chain of blocks containing validated transactions, across the decentralized network.

Blockchain nodes regularly reach a consensus about the authenticity of new transactions collected in blocks by solving computationally expensive cryptographic hashes, in other words, “Proof of Work”, and the fastest node to solve the puzzle is rewarded with an incentive. Over recent years, blockchain has been leveraged in various areas, most commonly concerning cryptocurrencies and financial services and of course many other application areas exist such as asset management, records management, identification, attestation, charity, government, education, health, to name only a few.

2.2 Social Networking Sites (SNS) and Shared Economy

The history of the Web dates back to 1989 with the invention of the Internet followed by the successful communication between a web browser and server a year later, thus

marking the beginning of the age of the Web. By 1999, there were around 3 million websites, but these were mostly static, read-only sites [2]. This generation of web or Web 1.0, saw unidirectional information flow from producer to consumer, via search engines and the most basic form of online shopping carts [3]. The need to engage and contribute actively led to the emergence of Web 2.0, the read-write-publish Internet that enabled users to share, collaborate and create content through social media and blogging. This age witnessed the birth of SNS such as Facebook and Wikipedia. Web 3.0 consisting of semantic markup and web services, facilitated machine-machine interaction while Web 4.0 is envisaged as the Internet of Things (IoT).

We have reached what “is also known as the symbiotic web. The goal of the symbiotic web is interaction between humans and machines in symbiosis. The line between human and device will blur. Web 4.0 will interact with users in the same way that humans communicate with each other. A Web 4.0 environ will be an “always on,” connected world. Users will be able to meet and interact inter-spatially on the web through the use of avatars” [2].

This gradual overall evolution of the Web caused much progression of the social web; the enormous development of technologies, particularly mobile devices, paved the way for the SNS to become pervasive. Tim-Berners Lee’s vision of the web as a collective intelligence is turning out to be true when considered in the light of the advancements of SNS [4]. Some instances of the most popular SNS other than Facebook and Twitter are Wikipedia that “harnesses collective intelligence to co-create content”, Flickr and Instagram that enable free photo sharing, YouTube that enables free video sharing, Delicious that facilitates discovering, storing and sharing of web bookmarks, Digg that helps in sharing opinions on digital contents and visualizing the rank of those contents, LinkedIn connects people to form professional network and Blogs for reading, writing, sharing and discussing users’ thoughts and opinions. In SNS, therefore, the contents are not produced or consumed by the platform owners, but by the users of such platforms.

Thus, emerged the concept of the shared economy where the platform owners will not themselves own the resources shared through the SNS, rather they will only manage or govern the functioning of such platforms that will bring buyers and sellers under one roof. Uber and AirBnB are some of the notable examples of such applications that empowered people to monetize their assets by sharing with peers through a synergetic platform. The platforms incentivize revenues from the interaction between the collaborating peers. However, various factors are required to be considered thoroughly such as legalities, taxes, security and insurance before the shared economy can become more mainstream. SNSs are not free from security risks and privacy threats. The massive amount of personal data shared on SNSs exposes users to Internet threats like “identity theft, spamming, phishing, online predators, Internet fraud, and other cybercriminal attacks” [5].

Of the various security and privacy issues discussed in [5], Rathore et al. discusses an interesting form of hazard, Sybil attack where the attackers generate a large number of fake identities facilitating them to gain advantages in the distributed and peer to peer system. SNSs may suffer from Sybil attack as they incorporate a plethora of users who joined as peers in a peer to peer network. In such environments, a single online entity can have multiple fake identities where attackers can outvote legitimate users, “reduce

reputation values, corrupt information”, vote an SNS account as the “best” increasing the account’s reputation and popularity.

2.3 Impact of Blockchain on Social Networking

As blockchain built upon provenance, immutability and distributed consensus, is fashioning the decentralized peer to peer communication process, some technologists are of the opinion that it can resolve the current challenges existing with SNSs [6]. In the absence of any centralized body governing user’s activities, the latter can benefit from greater security with blockchain. Raviv [8] highlights some of the advantages of using blockchain for social networking.

The User is not the “Product”

Most of the SNSs utilizes user information, including text, images, videos, available in their platform for advertising and marketing purposes. But in a blockchain-based social network, in absence of any platform owner, the nodes participating in the validation process can get rewards from the cryptocurrency bolstering such platforms. So, the user generated content may not be employed for analytics or marketing.

Enhanced Authority of Users over their Content

In the absence of centralized administration, there is no single entity controlling the content produced by users.

Better Security

Though the major SNSs have end-to-end encryption, the metadata exchanged with the messages can be eavesdropped by third parties. Blockchain’s decentralized and distributed ledger technology in addition to securing user data through encryption enhances user’s privacy and anonymity [7].

Censorship Resistant Applications

Blockchain-based SNSs ensure secure authentication as well as user anonymity, which is of great importance in censorship critical applications. “Platforms like Obsidian provide a way for users to circumvent the censors and to avoid surveillance” [8]. The decentralized library of Alexandria is yet another example of dissemination of art and information without censorship or ads [13].

Support for Payment Mechanism

While traditional SNSs like Facebook supports financial transactions, blockchain has the potential to revolutionize the shared economy in a peer to peer ecosystem. Blockchain was invented to support the generation of cryptocurrency and hence, the exchange of tokens or coins in a peer to peer social network is inherently accommodated by blockchain. Smart contracts can assist users perform deals through contracts signed and executed cryptographically.

Crowdfunding Possibilities

Blockchain-based SNS can help users in raising money through crowdfunding. Such networks can also expedite crowd-sales.

2.4 Existence Blockchain-Based Social Networking Applications

Irrespective of the issues persisting with the blockchain technology such as scalability, blockchain-based social networking and social media applications are multiplying. Researchers such as Qin et al. are already investigating the possibilities of utilizing blockchain in academic social networks supporting “irrevocable peer review records, traceability of reputation building and appropriate incentives for content contribution” [11]. Yet another example of blockchain-based social media is Ushare that allows users to control, trace and claim ownership of the contents they share over the decentralized, peer to peer, secure, anonymous and traceable content distribution network [12]. Here, we discuss some of the blockchain-based SNSs.

Sapien

This is a decentralized reputation based social news platform that takes into consideration four fundamental values: Democracy, Privacy, Freedom of Speech and Customizability. It has its own token, SPN and rewards the digital content creators without any central authority. This platform provides users with the flexibility to participate anonymously or use their real identity [10].

Sapien supports public or private communities called branches pertaining to specifics topic or themes. Users can participate in one or more branches and have reputations for each of those branches signifying their importance. It uses a Proof of Value Protocol. In essence, SPN supports collaborative distinguishing of high-quality content across the Sapien Network and this adds value to SPN. The protocol will bring forward quality content and reward users accordingly.

User's contents are evaluated across the network generating a score that designate user's reputation thus reflecting her domain specific skill within her communities. Thus, “within Sapien, reputation will mitigate trolling and reduce the spread of fake news” [10]. Sapien also enhances interaction among users by providing facilities such as friend addition, group creation, post sharing, comments writing and text and voice chat. Online privacy is protected through encrypted chat conversations. Sapien follows a hybrid centralized-decentralized web application during the initial stages with the social platform being centralized and all SPN transactions are decentralized and handled by the Ethereum Blockchain.

Steemit

Based on a decentralized network, Steemit aims to build communities and promote social interaction incentivizing users with cryptocurrency rewards in return of their posts depending on their upvotes count. Some of the services made available to the users of the Steem community are:

- Access to organized news and commentary
- Access to suitable answers to customized questions
- “A stable cryptocurrency pegged to the US dollar”
- Payments without charges [9]

It has an internal cryptocurrency token called STEEM that supports one-STEEM, one-vote. So, the greatest contributing users determined by their account balance will

have the strongest say about how the contributions are scored. Users have “financial incentive to vote in a way that maximizes the long-term value of their STEEM” [9]. It supports micropayments for users’ services. Users can upvote or downvote contents and rewards for contributors will be decided based on users’ votes. The system promotes the use of blockchain for censorship resistant applications. Here, no individual can censor contents which the STEEM possessors have valued.

Indorse

Based on the Ethereum blockchain, this is similar to LinkedIn. It has its own economy and currency. Users can participate in content creation reserving their rights, make profiles, connect with other users and receive payments for their participation. Indorse supports a “skills economy” leveraging Indorse Rewards, Indorse Score Reputation system and payment of tokens to users for their contributions in constructing the platform. User motivation is further enhanced through improvement of security and privacy for users’ data and information. It also leverages smart contracts “and pairs them with payment channels” [7].

Obsidian Messenger

This enables data storage in a decentralized system utilizing Proof of Stake consensus protocol and does not use it for analytics or advertising. An end-to-end encryption following 256 Bit encryption algorithm is applied to messages, files, pictures, videos and payments. The nodes receive cryptocurrency to enforce reliability, resiliency, and availability of the network. User accounts are not required and “the address will never have information that links to phone numbers, email or other accounts” [7].

Nexus

This supports charity, crowdfunding, transactions in a marketplace, purchasing ad space and other operations through cryptographically-signed and executed contracts. As a blockchain-based system, this application has internal currency thus bypassing the need for external payment mechanisms. This takes into consideration the requirement for secured authentication and anonymity in restrictive domains [7].

Others

There are many more SNS applications that utilize blockchain such as Sola, onG.Social, PROPS Project, Synereo, Golos, Akasha, Alexandria and SocialX.

3 The Challenge of Cash Currency Exchange

Now that we have seen how blockchain can contribute to building SNS and shared economy applications efficiently, we study a particular case of exchanging cash currency leftover from foreign trips. We explore how blockchain and social networking can be utilized together to solve the issue of cash-based leftover foreign currency exchange at the end of an international trip.

Almost every traveler possesses some amount of cash foreign currencies on her return from an international trip. But there is no mechanism to dispose of these currencies

conveniently and profitably. A huge amount of money that can add up to billions globally, remains unused in most of the travelers' wallets and cupboard drawers at home. Exchange through banks or other financial institution requires travelers to personally visit the institution to deposit the cash when the exchange rate is suitable and hence, is not convenient. Besides, the exchange rate is decided by the institution. It also charges additional fees, thus not lucrative for exchanging small amounts. Not all institutions accept coins. Some institutions have started exchanging cash currencies via posts or kiosks, however, they too suffer from the issue of centrally decided exchange rates, traveler's onus of submitting the cash currencies, exchange rate is based on the day when institution received the currencies, and limitation in terms of the list of currencies accepted for exchange.

Recently, a few systems have emerged that allow peer to peer exchange of currencies but they do not support exchange of cash currencies. Therefore, to address these pertinent challenges, we devised e-Mudra, an application facilitating peer to peer currency exchange, based on blockchain.

4 Method

We now describe e-Mudra, a blockchain-based Social Networking application connecting currency exchangers.

4.1 The Application Functionality

Our e-Mudra blockchain application maintains an online community of travelers who can exchange or transfer currencies among themselves. Any traveler must deposit cash foreign currency at a designated kiosk at the end of a foreign trip which gets added to her currency balance in her multicurrency wallet. She can then publish an advertisement for foreign currency exchange with her preferred exchange rate based on her currency balances and if any other traveler finds her published advertisement suitable, that traveler can initiate an exchange. Alternatively, she can choose to exchange currency with a fellow traveler if she finds the latter's advertisement for currency exchange profitable and matching her requirement. Travelers can exchange currencies, transfer currencies to other peers, donate currencies to charitable institutions, buy gifts with the currency balances in their multicurrency accounts and send them to other peers anytime once they have submitted the cash at the kiosk.

Every user has a username in addition to the public key that she can share with other peers in the network. Peers can transfer funds or send gifts to other peers using their usernames instead of using the complex public keys.

4.2 Scenario

The user deposits her cash currency, say X, in a smart kiosk placed at a convenient location such as airport before travelling back to her own country. The kiosk accepts the currency X and sends a receipt to her smartphone. The user later checks her multi-currency account in e-Mudra application which shows her the updated balance for currency X. She can

later post an advertisement for a target currency, say Y, in return of a part or whole of the balance amount of currency X, with a chosen exchange rate.

If any peer accepts the exchange at the user's published rate, the exchange takes place and the user is notified. Alternatively, the user instead of publishing her own rate, may select from any of the advertised rates from other peers in the network, requesting for an amount of currency X in return of currency Y. Once exchanged to currency Y, the money can be sent to the user's external account pertaining to currency Y or collected from a kiosk that operates for currency Y.

4.3 Architecture

Essentially, the users of this system are travelers and they cannot be expected to do the block validation unlike the Bitcoin blockchain network where any node can participate in consensus mechanism. Therefore, architecturally, e-Mudra leverages a permissioned consortium blockchain such that a selected set of nodes with known identities owned by stakeholder organizations that perform block addition and validation. The stakeholder organization is likely to be the one that owns the currency deposit and withdrawal kiosks. The network comprises of the following types of nodes, see Fig. 1.

- **The user nodes or clients that perform transactions** – these are the computer applications, i.e. the multicurrency wallets, which the travelers use to exchange funds, send money to peers, donate to charity or buy and send gifts.
- **The gateway nodes that conform to consensus protocol** – these nodes owned by the stakeholder organizations perform addition of blocks and validation of transactions.

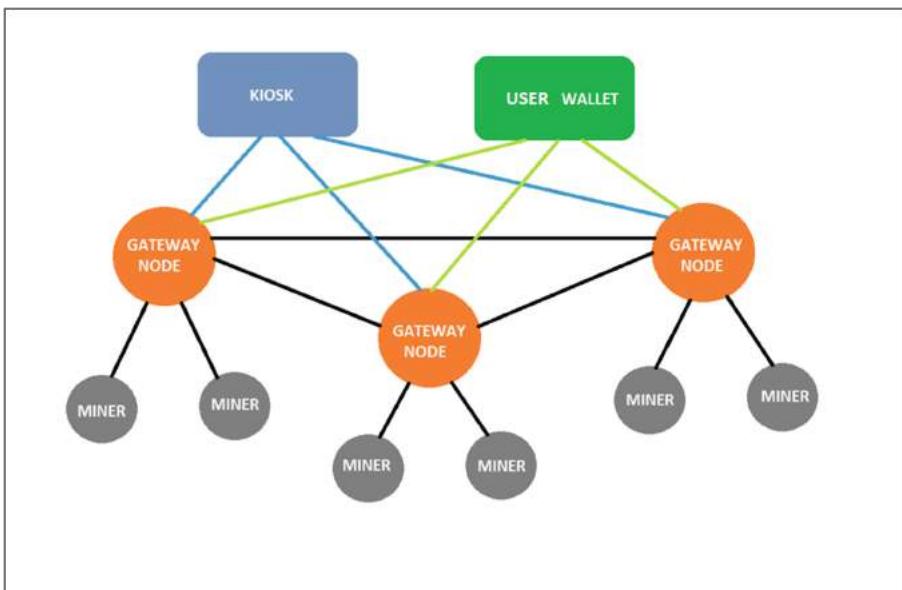


Fig. 1. The eMudra network

The gateway nodes from different organizations compete to find the next block and win rewards following the Proof of Work consensus protocol. The gateway nodes are connected to each other.

- **The miners that assist the gateway nodes in block mining** – each gateway node is connected to a set of miner nodes which help the gateway nodes in speedy discovery of next block by providing faster solution of the cryptographic puzzle.
- **The kiosk nodes that support cash deposit or withdrawal transactions** – these are the kiosks where users can deposit or withdraw foreign currencies.

The gateway nodes, miners as well as the kiosk nodes will be managed by cross-country organizations regulating the network.

A kiosk is connected to all the gateway nodes. When a traveler deposits or withdraws money at a kiosk, the corresponding transaction is broadcasted to all the connected gateway nodes. Similarly, when a traveler exchanges or transfers money through her multicurrency wallet, the corresponding transaction is broadcasted to all the connected gateway nodes. The gateway nodes collect the transactions and gather them in a transaction pool. When the number of transactions in the pool exceeds a threshold value, the gateway nodes collect the transactions from the pool and conduct mining. When a gateway node solves the cryptographic puzzle and finds a new block, it broadcasts it to other gateway nodes as well as its helper miners.

When a gateway node receives a new transaction, it sends the transaction to all its miners. If any of the miner finds a new block before the connected gateway node, it sends it to its connected gateway node which broadcasts the block to its other helper miners as well as other gateway nodes in the network and the gateway nodes and their miners then check if the block is a valid one. If it is a valid block, they add it to their local blockchain. If the gateway node finds a block before any of its helper miners, it forwards it to its helper miners as well as other gateway nodes in the network which check the validity of the new block and if the block is a valid one, they add it to their local blockchain. When a gateway node or a miner finds a new block, it receives 100 mudras, the internal cryptocurrency of the system.

All transactions are recorded on a shared distributed ledger such that any gateway node can read the transactions. A username is generated at the time of user registration and mapped against the user's public key.

A traveler must publish an advertisement or select from a list of advertisements to exchange currencies. When a traveler publishes an advertisement, it gets broadcasted from the traveler's multicurrency wallet to all the gateway nodes. Its default status is Active. When two travelers exchange currency, the corresponding advertisement status is updated as Expired. The advertisements are stored by the gateway nodes that manages the list of advertisements published by the travellers in the network.

5 Results

The differences of this application with existing systems are:

- The traveler's leftover foreign currency does not need to be exchanged on the same day when the currency is deposited or received by the governing institution(s).

- The user can exchange currency on any day post submission of the cash currency when she agrees on a profitable rate.
- The exchange rate is not centrally decided but only regulated to prevent money laundering; the user selects the preferred rate.
- Any currency of any denomination is acceptable as long as the user deposits them in the kiosk before leaving the respective country.
- The users can make good use of even low value cash foreign currencies not suitable for utilization through other mediums except charity.
- There is no exchange or transfer fees as the application aims to make small money exchange or transfer profitable, however there will be a subscription fee to use the application.

It is beyond the scope of this paper to detail the code base; however the proof-of-concept code base is available on GitHub on request.

6 Conclusion

Blockchain is a revolutionary disruptive technology that has much to offer in the domain of social networking. This paper has projected a brief overview of how social networking apps can benefit from blockchain and cited some relevant examples to corroborate the view. It further demonstrated a particular problem, that is, cash leftover foreign currency exchange being addressed by a blockchain-based social networking platform, e-Mudra that we proposed. While such a system as e-Mudra is complex and involves other innovative concepts like smart kiosks or internet of things, blockchain plays a fundamental role in enabling cash leftover foreign currency exchange profitably and conveniently. The current ‘proof of concept’ Java code base is available for collaboration on request to the principle author.

6.1 Future Work

Currently, the application prototype supports transfer and exchange of currencies, donation can also be performed by transfer functionality. However, buying and sending gifts is not supported by the prototype at the moment. Kiosk functionality is simulated. To transfer money, a user needs to know the username of the recipient which can be directly communicated by the recipient to the user. To add flavors of SNS, provision to add known users to a peer’s contact list can be supported. User can then add users whose usernames are shared with her directly to her contact list, update and delete later the same. She can send or exchange money with the users in her contact list and also buy gifts for them when the application will automatically fetch details of the recipient such as delivery address from the system and the sender will not need to provide the same.

References

1. Nakamoto, S.: Bitcoin: A Peer-to-Peer Electronic Cash System. www.bitcoin.org: https://bitcoin.org/bitcoin.pdf. Accessed 7 August 2018

2. Letts, S.: What is Web 4.0? stephenletts.wordpress.com: <https://stephenletts.wordpress.com/web-4-0/>. Accessed 7 September 2018
3. Web 1.0 vs Web 2.0 vs Web 3.0 vs Web 4.0 vs Web 5.0 – A bird's eye on the evolution and definition. flatworldbusiness.wordpress.com: <https://flatworldbusiness.wordpress.com/flat-edu-cation/Previously/web-1-0-vs-web-2-0-vs-web-3-0-a-bird-eye-on-the-definition/>. Accessed 7 September 2018
4. Rana, J., Kristiansson, J., Hallberg, J., Synnes, K.: Challenges for mobile social networking applications. In: Mahmood, R., Cerqueira, E., Piesiewicz, R., Chlamtac, I. (eds.) Communications Infrastructure. Systems and Applications in Europe. EuropeComm 2009. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, vol. 16. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-11284-3_28
5. Rathore, S., Sharma, P.K.: Social network security: issues, challenges, threats, and solutions. Inf. Sci. **421**, 43–69 (2017)
6. Social networking over blockchain, 29 March 2018. <https://www.scalablockchain.com: https://www.scalablockchain.com/blog/social-networking-over-blockchain/>. Accessed 7 October 2018
7. Kariuki, D.: Five Top Blockchain-Based Social Networks, 12 September 2017. <https://www.cryptomorrow.com: https://www.cryptomorrow.com/2017/12/09/blockchain-based-social-media/>. Accessed 7 October 2018
8. Raviv, P.: 6 Ways the Blockchain is Revitalizing Social Networking, 1 May 2018. <https://cryptopotato.com: https://cryptopotato.com/6-ways-blockchain-revitalizing-social-networking/>. Accessed 7 October 2018
9. steemit (June 2018). <https://steemit.com/: https://steem.io/SteemWhitePaper.pdf>. Accessed 21 July 2018
10. Ankit Bhatia, R.G.: sapien Decentralized Social News Platform (March 2018). https://www.sapien.network: https://www.sapien.network/static/pdf/SPNv1_3.pdf. Accessed 12 July 2018
11. Qin, D., Wang, C., Jiang, Y.: RPchain: a blockchain-based academic social networking service for credible reputation building. In: Chen, S., Wang, H., Zhang, L.J. (eds.) Blockchain – ICBC 2018. ICBC 2018. Lecture Notes in Computer Science, vol. 10974. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94478-4_13
12. Antorweep Chakravorty, C.R.: Ushare: user controlled social media based on blockchain. In: ACM IMCOM. Beppu, Japan (2017)
13. Alexandria. <https://www.alexandria.io: https://www.alexandria.io/learn/#learn1>. Accessed 20 July 2018



Real and Virtual Token Economy Applied to Games: A Comparative Study Between Cryptocurrencies

Isabela Ruiz Roque da Silva^(✉) and Nizam Omar

Mackenzie Presbyterian University, São Paulo, Brazil

Abstract. Monetization is a way to earn money from products, softwares or services. Many video game companies use monetization to sell items, bonus or even money to customize the user experience or achieve quests easily in their games. Even though the cryptocurrencies are becoming more and more popular, few games use cryptocurrencies as a real form of monetization, which means that there is a potential for its application. The purpose of this paper is to compare the most used cryptocurrencies for games, based on some important characteristics, discuss which is the most suitable cryptocurrency for gaming platforms and propose an architecture for gaming companies use inside the games using cryptocurrencies.

Keywords: Cryptocurrency · Blockchain in games · Game monetization

1 Introduction

According to Mishkin, 2015, monetization is the name given to the process of exchanging anything: a product, service or software, for a specific value that can be used to legally buy and sell goods. In the universe of video games, monetization is used to sell games and additional content to increase the custom user experience [5]. Every gamer has a different experience, making it necessary to adapt the game to the most suitable monetization, according to the user profile. More and more games are using custom monetizations to generate revenue and increase the way the player interacts with the game [3].

Since the beginning of Massively Multiplayer Online Games (MMO), the idea of virtual coins was discussed, for example, in the game *Everquest from Verant Interactive*. Inside *Everquest*, the users populated a world named Norrath, where they could create an avatar and live their virtual lives, using the game's currency; (Castranova, 2001) studied in details the social game mechanics and the macroeconomy of buying and selling items inside and outside the game (which was illegal from the company perspective). He concluded that inside the game, there was a supply and demand law on rare items: when a rare item was in high demand, its price rises in avatar-to-avatar market [4].

Based on the context, the cryptocurrencies can co-exist with electronic games due to using them will make the transactions faster and trackable. The idea of this paper is to present in details how the games monetize their activities nowadays and to propose an architecture to explain how cryptocurrencies could be implemented in well-known games to increase security, tracking and monetization of anything inside the game. Besides, discuss some cryptocurrencies that are being used on games based on their technology and market cap, comparing similarities and differences between them. In Sect. 2, concepts and related studies about monetization in games are presented; in Subsect. 3 some concepts about crypto-games and applications are raised; in Subsect. 4, the authors present a comparative study between all gathered cryptocurrencies in the raised crypto-games and in the last section, the proposed architecture for cryptocurrencies and games is proposed.

2 Monetization

At the beginning of game's history, the games were created by laboratories or computer scientists at universities. Then, the arcades games came to life where players could exchange money for coins to play on machines. They were formerly sold on floppy disks at small hardware specific stores, then the evolution came after some years and soon the video games were created, in which games were sold by physical copies [7], the most used method to monetize games until today.

Monetization in games is one of the most important phases of designing a video game. It is becoming more and more important for publishers to earn revenue from their games, most of them have been using a “free-to-play” approach where the users can download the game without paying anything, except if they want to customize their gaming experience by buying custom items, energy, weapons, etc. [6]: *Plants vs Zombies 2* allows the users to buy plant types to improve the gameplay and hence the score; in *Kingdom Rush Frontiers*, to level up easily, the user can buy power-ups to increase the character health and explode enemies [6]. Usually, the player does not buy these goods using cash or some kind of currency directly, the game has a proxy that connects to a third party financial company where the user can buy them using credit cards or another form of exchange [6] and that is where the cryptocurrencies could help the publishers. Despite mostly being a electronic game, some games sell physical goods to customize characters and improve gameplay mechanics, like the game *Disney Infinity* that sells physical toys representing new characters and environments [6].

Besides the “free-to-play” approach and the retail method to monetize games, there are many other forms of classifying monetization in this industry: David Perry, former CEO of a cloud gaming platform and Chief Creative Officer from *Acclaim Games*, in 2008 at the *Social Gaming Summit* mentioned 29 business models for game monetization based on the book “E-commerce: Business, Technology and Society” from Kenneth Laudon and Carol Traver [1, 2]; Tim Fields and Brandon Cotton in their book “Social Game Design” explain many ways

to generate revenue from games using three basic models: the Classic Download Model, the Signature Model and the Freemium Model [7]. Scott Rogers presented eight models of monetization: Trial; Freemium; Free-to-play (F2P); Downloadable Content (DLC); Season Pass DLC; Membership; Premium and Subscription [6]. Basically, all forms presented by different authors aim to earn money from different types of games and can be summarized into major types: Retail Purchase, In-game Microtransactions, Digital Download, Subscription Model and Indirect Monetization [8]. In this paper five types of models are presented: Trial Model; Retail Purchase; Digital Download; Subscription Model and Freemium Model.

In the Trial Model, the publisher launches a version of the game that is not complete, it is a sneak peek of how the entire game will be, including how the mechanics will work, the visual of the characters and the game's world. Players download it and can try the game without paying anything and the publisher hope they will pay for the full version [6]. This is a common way to attract players on consoles like *Playstation* [11] and *Xbox* [12]. For example, the games *Final Fantasy VII Remake* and *Far Cry 4* allow the players to play a limited time on the game (demo play).

The Retail Purchase Monetization is the most conventional way of monetizing games, the users pay for the physical copy of the game they want, like a CD and are ready to play the game [8]. It is a model that attracts few people each year due to the digital version of the games that sometimes is more affordable. That change of mind can be seen on the release of the new *Playstation 5*, where the player can choose between buying a more expensive version of the console with CD input or a low-priced version only for digital downloads.

In the Digital Download Model or Classic Download Model, the publisher makes an advertising campaign about the game that is being created, captivating and engaging the players to download the game and play it. Some examples are the mobile stores for phones and tablets [10,14]; the *Steam* website [15] containing a plenty of games with different characteristics, that diversify the target audience and the digital stores from consoles like *Sony Playstation* [11], *Microsoft Xbox* [12] and *Nintendo Switch* [13].

In the Subscription Model, there is a concept about game time. Instead of buying the entire game and additional packages as soon as the publisher launches some additional content, the users pay a monthly, quarterly, annual subscription to pay or only pay the time they will play [7]. The majority of games that use this kind of model are MMORPGs (Massively Multiplayer Online Role-Playing Game), like *World of Warcraft* from *Blizzard*, launched in November 2004, which can use the Freemium Model (to download only the base game) and Subscription Model for buying a specific period of time. Besides, the players can also buy mounts and mascots to customize the gaming experience and help them inside the game. Figure 1 and 2 show these two cases. Despite being an interesting way of monetizing a game, the real challenge of using it is to keep the player interested on the content of the game as the time goes by.



Fig. 1. Monetization using game time on *World of Warcraft* game

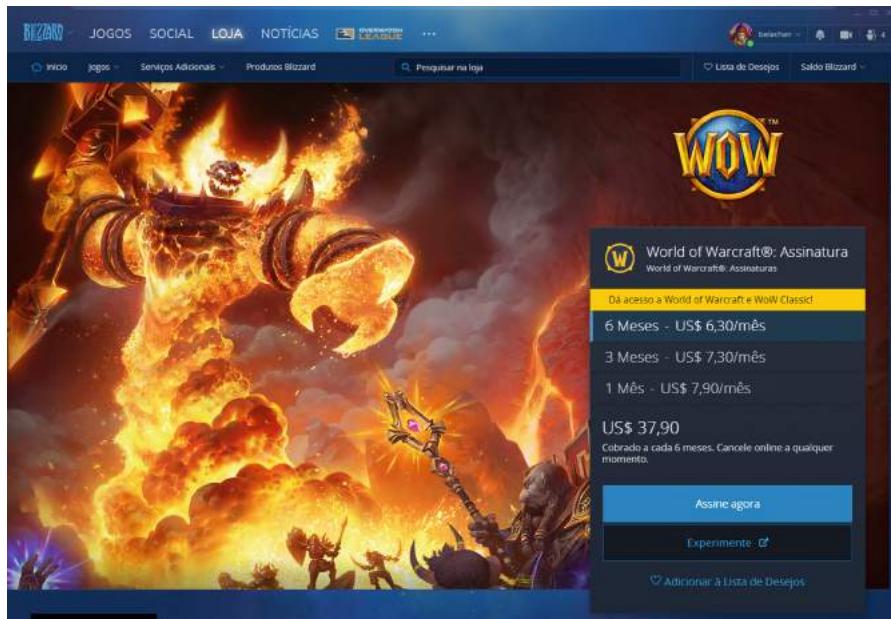


Fig. 2. Monetization using the subscription model on *World of Warcraft* game.

In the Freemium Model or Free-to-Play, the users do not pay for downloading the game [6, 7]. One of the greatest example of this kind of model is the game *FarmVille* from the *Zynga* company. The players could play for free, but could speed up some boring tasks that need friends to complete or time to wait. The publisher encourages the players to pay for time, virtual goods, locked contents and DLCs (Downloadable Content). It is becoming more and more popular, specially on mobile games.

Despite being completely distinct models, some games can benefit from more than one monetization model to specific parts [6, 7]. The *World of Warcraft* game is a fine example: beyond the subscription monetization, the player can also buy mounts and pets.

In 2009, *Facebook* created a system called *Facebook Credit* so all the games inside his platform could use a common currency, with the idea to earn a percentage on the revenue of the games and also increase security on user's transactions. A few developers were against the idea due to the *Facebook*'s fee to do that: almost 30% of the transaction [7], on the other hand, the community as a whole accepted the idea well. One of the benefits of *Facebook*'s system is to give friends a gift with credits to spend in any game they want [7].

After the creation of Blockchain and the cryptocurrencies, some games using the Blockchain's technology appeared and attracted attention to how this type of technology can impact the industry of games. The next section will discuss more about it.

3 Crypto Games

From the very beginning of the history of cryptocurrencies, there are ideas on how to use them for games. Some examples are: *Dragon's Tale*; *MinecraftCC*; *Gambit*. In 2010, the most famous indie game to use Bitcoin was called *Dragon's Tale*: a mix of MMO with casino, which had several activities based on four categories: Luck; Skill; PvP and Tournament [17]. For example, the Luck category has an activity called Palace Garden, in which players dig eight holes in the Chinese imperial garden. Each hole may or may not contain Bitcoins and when it finds two or three holes that have Bitcoins, the number increases.

In 2012 the first server to run Bitcoin was created along with *Minecraft*, the so-called *MinecraftCC*, in which players earned fractions of Bitcoins in exchange for building blocks and killing monsters. Despite community support and advertisements to maintain the project, in 2016 it was discontinued. After a while, in 2013, *Gambit* appeared, in which players could bet against each other in classic board games or cards using Bitcoin, it functioned as a form of virtual casino. As can be seen, in the early history of cryptocurrencies, almost all applications for games were some kind of virtual casinos or gambling applications.

Following this trend, a cryptocurrency appeared with this goal in 2014: DigiByte, which runs on top of the Blockchain technology [16]. The idea was to establish a connection between digital games and the tokens generated by the cryptocurrency. Even with all the effort to create this environment, projects using DigiByte were paused in 2017.

A year after the creation of the DigiByte cryptocurrency, *Blizzard* created an internal *World of Warcraft* game token for players to exchange their real coins for virtual gold so they could spend or sell at auction houses. Security has been improved, because if a player bought tokens at the auction house, he could use it only after thirty days, which helped prevent fraudulent accounts [18].

Looking at the potential of game monetization and all the applications above using cryptocurrencies, some companies developed researches allying cryptocurrencies/Blockchain with games, as it can be seen on Table 1. The Blockchain Game Alliance was founded in 2018 with the main aim of spreading crypto games and exploring the creation of games with distributed ledger technologies.

Table 1. Table with some cryptocurrency patents in the gaming industry. Taken from Google Patents (<https://patents.google.com/>).

Patent number	Patent name	Publication Year
TW201922325A	Blockchain gaming system	2018
US20190303960A1	System and method for cryptocurrency generation and distribution	2019
US20180114405A1	Digital currency in a gaming system	2019
US20190122495A1	Online gaming platform integrated with multiple virtual currencies	2019
US20180197172A1	Decentralized competitive arbitration using digital ledgering	2018
US9997023B2	System and method of managing user accounts to track outcomes of real world wagers revealed to users	2019

Analyzing all the history of gaming application with cryptocurrency, that is how the Crypto Games were born. Crypto Games is the name given to every game that uses distributed ledgers to operate the game and a cryptocurrency for exchanging items or characters for money.

There are many benefits of using blockchain in games, some researchers have already raised them [19, 20]; the rules of blockchain games are transparent, everyone can see what the game is about and what the player can do or not; it guarantees the ownership of items, characters or whatever element that the player owns inside the game [19]; with this guarantee, the owner of them can reuse these elements in other games inside the same Blockchain, like *CryptoKitties* and *KittyRace* [19]. *KittyRace* reuses some elements of the *CryptoKitties* game, so you can play both games with the same account [19].

Although it is a good strategy to use blockchain for designing games, [19] raised some restrictions about the visual design of the game due to technical limitations. It is a promising application but huge companies are still at an advantage on this aspect.

4 Cryptocurrencies in Games

Nowadays there are a large variety of crypto games on the internet for everybody who wants to play: they goes from gambling games to RPG games (Role-Play Games) [20]. Exploring deeper the cryptocurrencies atop of the most popular games to date, the authors present a table (Table 2) with some of these games and the cryptocurrencies they use as a base for transactions.

Table 2. Popular crypto games and their respective cryptocurrency.

Name of the game	Cryptocurrency
CryptoKitties	Ethereum
KittyRace	Ethereum
Satoshi dice	Bitcoin
TronBet	TRON
EOS knights	EOS
0xUniverse	Ethereum

As seen in the table above, there are four main cryptocurrencies used in games: Ethereum, Bitcoin, TRON and EOS. Bitcoin was the first cryptocurrency created and yet is used in various gambling games [17]. Although the market capitalization of the bitcoin is the biggest one with \$ 96 billion dollars, according to Coinbase platform, Ethereum has a lot of potential since the majority of games operates atop of it. One of the reasons why Ethereum (second-largest cryptocurrency by market capitalization) is the most used in games is because of its functionality of smart contracts and the DApp infrastructure to build a game or a application on it easily. DApps are becoming popular on the blockchain world since anyone can make an app with an interface to interact with the user in any programming language and contact the server in which the blockchain is running [9]. Another advantage of using Ethereum instead of Bitcoin is the transactions time: Ethereum can handle more transactions per time than Bitcoin, it also can support any type of computation and the creator of the DApp can make his own rules [24, 25].

Other cryptocurrencies like TRON and EOS also use the concept of DApp application. An annual report made by Dapp.com informed that TRON is the second most used DApp, losing only to Ethereum, which represents the majority of gaming platforms [26]. In Table 3 the authors bring a brief summary of the

Table 3. Dapp market summary in 2019 with Ethereum, EOS, Steem and TRON [26].

	All	ETH	EOS	Steem	TRON
Total number of dapps	2,989	1,822	493	92	520
Active dapps	2,217	1,129	479	80	482
Active users	3,117,086	1,427,093	518,884	120,560	967,775
Transactions	3.26B	24.52M	2.81B	85.72M	290.28M

DApp market in 2019 by Blockchain, emphasizing the number of DApps on Ethereum.

Analysing this data and the proposal of each cryptocurrency whitepaper [27–30], the authors come up with four main important characteristics for a gaming cryptocurrency: Transaction Fees (if the cryptocurrency has fees to transact); Smart Contracts (whether it has the possibility of implementing a smart contract to make the game smarter); Scalable (it refers to the ability of the cryptocurrency to scale in terms of numbers of transactions) and Performance (if the decentralized network bears many transactions per time). The results of rising these characteristics are on the Table 4.

Table 4. Comparative between the main important characteristics raised by the authors and the cryptocurrency.

Cryptocurrency	Transaction Fees	Smart Contract	Scalable	Performance
Bitcoin	X			
Ethereum	X	X		
TRON	X	X	X	X
EOS		X	X	X

From the scenario raised above, the cryptocurrency that is most suitable for games is EOS. It does not have any fees to play, so the player does not need to transfer funds to the account to make any small transaction, it scales and performs better than the others, since uses a Proof-of-Stake consensus algorithm, differing from the others that uses Proof-of-Work (Bitcoin [27] and Ethereum [29]), Delegated Proof-of-Stake (TRON) [28].

Besides EOS, TRON can also be another candidate for gaming as it is growing in Dapp applications in 2019. The number of active Dapps in EOS are almost the same as TRON Dapps, this comes from the fact that these two cryptocurrencies are mostly used for casinos platforms that involves gambling, Table 5 illustrates the market dominance by category on a daily basis.

Even though Ethereum is not fully scalable, it represents the majority of dominance in gaming category (28%), followed by EOS (15%), TRON (12%)

Table 5. Daily market dominance by category [26].

	All (%)	ETH (%)	EOS (%)	Steem (%)	TRON (%)
Game	22	28	15	12	12
Gambling	27	20	56	7	37
High-risk	21	24	4	0	38
Exchange	4	5	8	3	4
Finance	4	5	2	3	1
Social	6	3	3	52	2
Art	2	3	1	1	0
Tools	7	5	4	18	2
Others	8	7	10	4	4

and Steem (12%). Since Ethereum came before the others, this may be the reason why developers prefer using it, for security concerns.

5 Proposed Architecture

This section provides some problems that cryptocurrencies could solve in games, along with a explanation about the proposed architecture and strengths of using it.

The video game industry is one of the biggest industries in the world. According to *New Zoo*, a research company about game market insights, the global games market has a market cap about \$159.3 Bi only on 2020, which is a growth of 9.3% compared to last year [21].

As a well-known fact, the decentralized applications, the blockchain technology and its cryptocurrencies represents a world where third-party companies will not have much influence on markets. Instead of using the services from this companies, like credit card, the video game companies could benefit from services from blockchain and cryptocurrencies to ensure the security of the player's money and its personal data [19, 20], which means it will reduce the third party's partnership.

There are many problems on the actual finance structure of the games: first, the players need to trust their credit card data or bank account to a third-party company, which could lead to vulnerabilities and information leak if a hacker attacks the company, what has happened recently with some games [22, 23]. On 2020, hackers modified some game's database to generate items or virtual cash for their own account, so they could sell on a market [22]. During the coronavirus pandemic, these attacks are becoming more frequent, due to the fact that the players are often online and unsuspecting any attack. The attacks range from phishing to theft of sensitive information such as credit card data [23].

The proposed architecture of this paper could solve the theft of credit card information, since the player does not need to enter any credit card information, only the cryptocurrency address. Figure 3 presents our proposal of solution.

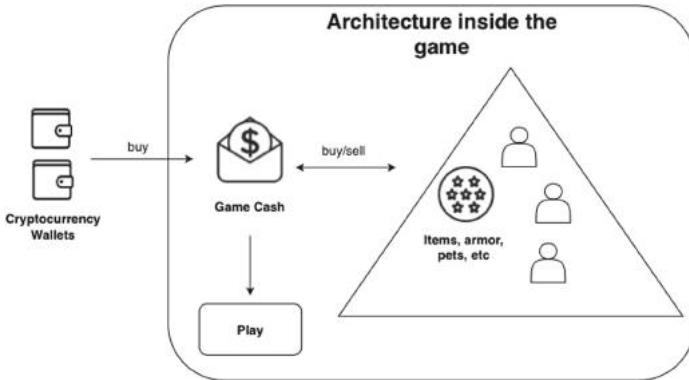


Fig. 3. Proposed architecture.

In the solution, the publisher of the game only needs the blockchain to guarantee the ownership of the items between players and secure the player's game cash. The player can buy the game cash using the address of his cryptocurrency wallet: he transfers funds between his wallet and the game's wallet. As appointed in the previous section, some cryptocurrencies that could be used on the game is EOS and TRON, due to their beneficial features and network hashrate. When the transaction is approved by the blockchain, the player has two options: buy or sell items, armor, etc. using his new balance from other players or just play the game. He could also receive his game cash back in form of cryptocurrency for his wallet.

This type of architecture can ensure the security by using blockchain for validating transactions: the hackers will not be able to attack the monetary system or the trading system, because of the immutable feature of blockchain. And also, they will not be able to steal the player's credit card information or his balance from the wallet, since the cryptocurrencies use Asymmetric Encryption Cryptography, Hashing and Consensus Algorithms to secure the wallet, the network and personal data of the users.

6 Conclusions and Future Work

Since the beginning of the game's industry, monetization is a way of earning money from the players. Each game use a different type of monetization or combine different types to achieve a better game experience.

In this paper the authors discussed about the importance of the crypto games and their benefits, as well as the different types of cryptocurrencies that could

be used in games based on some important features. The proposed architecture demonstrated what could be done in the financial structure of the game to prevent hacker attacks and information theft, which guarantees the integrity of the whole system and could be used in any games, even the ones that already exist.

For future work, the most suitable type of games for blockchain will be analysed, other cryptocurrencies not raised on this article will be compared and a further research DApp platforms for games will be done.

Acknowledgments. The authors thank Mackenzie Presbyterian Institute and Mack-Pesquisa for the financial help to develop this research.

References

1. Perry, D.: 29 Business Models for Games (2008). <https://lsvp.wordpress.com/2008/07/02/29-business-models-for-games/>. Accessed 25 Sept 2020
2. Laudon, K., Traver, C.: E-Commerce: Business, Technology and Society. Pearson (2008)
3. King, D., Delfabbro, P.: Video game monetization (e.g., ‘loot boxes’): a blueprint for practical social responsibility measures. *Int. J. Mental Health Addiction* (2018)
4. Castranova, E.: Virtual Worlds: A First-hand Account of Market and Society on the Cyberian Frontier. Gruter Institute Working Papers on Law, Economics, and Evolutionary Biology, vol. 2, December 2001
5. Mishkin, F.S.: The Economics of Money, Banking and Financial Markets. Prentice Hall (2015)
6. Rogers, S.: Level UP! The Guide to Great Video Game Design, 2nd edn. Wiley (2014)
7. Fields, T., Cotton, B.: Social Game Design. Elsevier (2012)
8. Fields, T.: Mobile & Social Game Design: Monetization Methods and Mechanics, 2nd edn. CRC Press (2014)
9. Cai, W., Wang, Z., Ernst, J.B., Hong, Z., Feng, C., Leung, V.C.M.: Decentralized applications: the blockchain-empowered software system. *IEEE Access* **6**, 53.019–53.033 (2018)
10. Google: Google Play Store. 2020. Accessed 01 Feb 2020. <https://play.google.com/store/apps>
11. Sony Playstation Store: Playstation Store (2020). Accessed 24 Sept 2020. <https://store.playstation.com/pt-br/home/games>
12. Microsoft Xbox Store (2020): Xbox Store. <https://www.xbox.com/pt-BR/games/all-games?xr=shellnav>. Accessed 24 Sept 2020
13. Nintendo Switch Store: Switch Store (2020). <https://www.nintendo.com/games/switch/>. Accessed 24 Sept 2020
14. Store, Apple: Apple App Store. 2020. <https://www.apple.com/ios/app-store/>. Accessed 24 Sept 2020
15. Steam: Steam Store (2020). <https://store.steampowered.com>. Accessed 01 Feb 2020
16. DigiByte: DigiByte Global Blockchain (2014). <https://www.digibyte.co/digibyte-global-blockchain>. Accessed 07 Oct 2019
17. eGENESIS. Gambit website (2010). <http://www.dragons.tl/>. Accessed 07 Oct 2019
18. Blizzard. World of warcraft token (2015). <https://us.shop.battle.net/en-us/product/world-of-warcraft-token>. Accessed 07 Oct 2019

19. Min, T., Wang, H., Guo, Y., Cai, W.: Blockchain games: a survey, June 2019
20. Scholten, O., Hughes, N., Deterding, S., Drachen, A., Walker, J., Zendle, D.: Ethereum crypto-games: mechanics, prevalence and gambling similarities, October 2019
21. Zoo, N.: New Zoo - Key Numbers. 2020. <https://newzoo.com/key-numbers/>. Accessed 27 Sept 2020
22. Magazine, PC: Feds charge 5 chinese hackers for targeting video game companies. <https://www.pcmag.com/news/feds-charge-5-chinese-hackers-for-targeting-video-game-companies>. Accessed 27 Sept 2020
23. Beat, Venture: Akamai: Cyberattacks against gamers spiked in the pandemic (2020). Accessed 27 Sept 2020. <https://venturebeat.com/2020/09/23/akamai-game-industry-faced-more-than-10-billion-cyberattacks-in-past-two-years/>
24. Vujicic, D., et al: Blockchain technology, bitcoin, and Ethereum: a brief overview. In: 2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH), pp. 1–6 (2018)
25. Rudlang, M.: Comparative analysis of bitcoin and ethereum. Master's thesis, NTNU (2017)
26. Dapp, R.: 2019 Annual Dapp Market Report. <https://www.dapp.com/article/dapp-com-2019-annual-dapp-market-report>. Accessed 01 Feb 2020
27. Nakamoto, S.: Bitcoin: A peer-to-peer electronic cash system. Cryptography Mailing list (2009). <https://metzdowd.com>
28. TRON.: Tron: Advanced decentralized blockchain platform. TRON Foundation, Technical report, December 2018
29. Buterin, V.: A next-generation smart contract and decentralized application platform. Technical report, May 2018
30. Grigg, I.: Eos - an introduction. Technical report, March 2018



Blockchain Smart Contracts Static Analysis for Software Assurance

Suzanna Schmeelk¹(✉), Bryan Rosado¹, and Paul E. Black²

¹ Computer Science, Mathematics and Science, St. John's University, New York, NY, USA

schmeels@stjohns.edu, bryan.rosado16@my.stjohns.edu

² National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA

paul.black@nist.gov

Abstract. This paper examines blockchain smart contract software assurance through the lens of static analysis. Smart contracts are immutable. Once they are deployed, it is impossible to patch or redevelop the smart contracts on active chains. This paper explores specific blockchain smart contract bugs to further understand categories of vulnerabilities for bug detection prior to smart contract deployment. Specifically, this work focuses on smart contract concerns in Solidity v0.6.2 which are unchecked by static analysis tools. Solidity, influenced by C++, Python and JavaScript, is designed to target the Ethereum Virtual Machine (EVM). Many, if not all, of the warnings we categorize are currently neither integrated into Solidity static analysis tools nor earlier versions of the Solidity compiler itself. Thus, the prospective bug detection lies entirely on smart contract developers and the Solidity compiler to determine if contracts potentially qualify for bugs, concerns, issues, and vulnerabilities. We aggregate and categorize these known concerns into categories and build a model for integrating the checking of these categories into a static analysis tool engine. The static analysis engine could be employed prior to deployment to improve smart contract software assurance. Finally, we connect our fault categories with other tools to show that our introduced categories are not yet considered during static analysis.

Keywords: Blockchain · Smart contracts · Solidity · Ethereum Virtual Machine (EVM) · Software Assurance · Static analysis

1 Introduction

Smart contracts are immutable. Once deployed, they cannot be changed. Modification to the smart contract requires the entire blockchain to be tossed aside, which is an infeasible solution to most production blockchains. This work extends the work of Dingman et al. [1] to closely examine different types of known bugs that can be introduced during smart contract development. The better the understanding of the secure software development for smart contracts on blockchains, the more robust the overall blockchain. In fact, development tools such as static analysis and testing can be further developed to identify software smart contract concerns. Arguably, all software faults can have some level of security ramifications—perhaps, minor. This research examines the Solidity v0.6.2

developer notes to classify categories of faults noted by developers. Some of these categories are not well understood and can serve as resources for further improvement of Solidity static analysis tools to detect smart contract development concerns.

2 Review of Literature and Related Work

The assurance of smart contracts pre-deployment is extremely important as they cannot be patched once executed on a chain. Currently there are four main research pillars with respect to smart contracts: use cases, correctness and verification, other software assurance methodologies and data assurance. Let us now examine the current literature in each subdomain.

2.1 Smart Contract Use Cases

Smart contracts are being applied in many industries. Juho et al. [30], Hao et. al [10], and Pee et al. [18] discuss block chain usage in the public sector. Afanasev, Fedosov, Krylova and Shorokhov [14] have reported on the integration of smart contracts with cyber-physical systems to improve the reliability, fault tolerance and security of machine-to-machine communications. For similar reasons, Christidis and Devetsikiotis [20] and Naidu, Mudliar, Naik and Bhavathankar [11] made a case for integrating the Internet of Things (IoT) architecture [20] and supply chain [11] with blockchain. Medical image research integrating smart contracts has been published by Tang, Tong and Ouyang [12]. Bringing smart contracts into the educational arena to verify diplomas, certifications, degrees, etc. has been reported by Cheng, Lee, Chi and Chen [13]. Governatori et al. [8] discuss legal concerns around smart contracts. With the advent of the European Union's General Data Protection Regulation (GDPR), smart contracts have been suggested by Neisse, Steri and Nai-Fovino [17] for the accountability and provenance for data tracking.

2.2 Smart Contract Correctness and Verification

Research has also explored the correctness and formal verification of smart contracts. Bartoletti and Zunino [15] introduced BitML (short for the Bitcoin Modelling Language) which is a calculus for bitcoin smart contracts to exchange currency according to contract terms. Lee, Park and Park [9] introduced a formal specification technique for verifications in smart contracts. And, Schrans, Eisenbach and Drossopoulou [16] propose a new type-safe, capabilities-secure, contract-oriented programming language, Flint, for future development.

2.3 Other Smart Contract Software Assurance Methodologies

Dingman et al. [1] examined Blockchain Smart Contract bugs through the lens of the National Institute of Standards and Technology (NIST) Bugs Framework [19]. Their work identifies nearly 50 bugs spanning multiple categories (e.g. Operational, Security, Functional, Developmental, etc.).

Tikhomirov, Voskresenskaya, Ivanitskiy, Takhaviev, Marchenko and Alexandrov [23] introduce a pattern-based static analysis tool, SmartCheck, developed in Java. SmartCheck translates Solidity programming language contracts into an XML-based intermediate representation. SmartCheck then checks the XML against XPath patterns. They report that they classify Solidity issues based on the work by Kevlin Henney [24]. Potential vulnerabilities identified by SmartCheck are given in a table, where rows highlighted in gray are susceptible to being false positives. In addition, their research identifies four domains of issues which can be identified with static analysis: security, functional, operational and development.

Luu, Chu, Olickel, Saxena, and Hobor [21] introduce Oyente, which is a symbolic execution tool to identify weaknesses. The authors also identified vulnerabilities within four main categories: transaction-ordering dependencies, timestamp dependencies, mishandled exceptions and re-entry issues.

Juels, Kosba, and Shi [5] report on the potential for malicious smart contracts, called criminal smart contracts, to be introduced into a blockchain environment. Such criminal smart contracts may attempt to steal private keys or attempt to facilitate crimes within the distributed smart contract systems.

A few papers have referenced known smart contract vulnerabilities. Wohrer and Zdun [22] report on known security vulnerabilities in Ethereum/Solidity smart contract systems. They classify the vulnerabilities into security patterns and reference an example of such a vulnerable contract as follows: balance limit, mutex, rate limit, speed bump, emergency stop and checks-effects-interaction. Mense and Flatscher [3] reported on three known classes of vulnerabilities (based on the work of Atzei, Bartoletti and Cimoli [25] and Dika [26]) in Solidity, the Ethereum Virtual Machine (EVM) and in the underlying blockchain framework.

Xu, Pautasso, Zhu, Lu and Weber [2] identified a set of patterns for blockchain applications which can be broken into components for further vulnerability scrutiny. Specifically, they reported on potential issues: external world interactions, data management, security and contract structural patterns.

2.4 Smart Contract Data Assurance

Another domain of smart contract research involves developing and protecting a blockchain architecture and methodology for smart contracts to retrieve and authenticate data from external sources. Bjorn van der Laan, Oğuzhan Ersøy and Zekeriya Erkin [7] propose a model concept around a blockchain oracle, entitled Multi-Source oraCLE (MUSCLE), to reliably retrieve data from outside systems for use within smart contracts.

2.5 Developing Smart Contracts

Terry [4, 6] discusses the software development process and basic concepts for smart contracts. Terry covers topics including: the fundamental idea of smart contract, the block chains relationship with smart contracts, and fundamental crypto currency exchange.

2.6 Smart Contract Attacks and Protection

Sayeed et al. [31] discuss classic smart contract attacks and mitigations. Their research shows the relationship of the classic Decentralized Autonomous Organization (DAO) attack and the Parity Wallet attack. Such attacks have cost organizations and people many millions of dollars. Specifically, they classify blockchain exploitation techniques into categories based on the attack rationale: attacking consensus protocols, bugs in the smart contract, malware running in the operating system, and fraudulent users.

3 Static Analysis to Detect Errors

Blockchain smart contracts are being integrated in many different domains across the world. One of the deployment difficulties is that smart contracts are immutable. Detecting smart contract errors and potential issues, which are prone to security concerns prior to deployment is essential. For this research we focused on v0.6.2 of the Solidity API [27]. Currently there are many warnings being posted about the development of smart contracts which are unchecked by the current Solidity compiler. Thus, the burden to detect these vulnerabilities and concerns are entirely on the Solidity developer. Our research contribution is to categorize many of these warnings at large for further integration into a static analysis engine to detect common programming concerns.

4 Findings

Specifically, this work focuses on smart contract concerns in Solidity v0.6.2 which are unchecked at either compile or deployment time. Solidity, influenced by C++, Python and JavaScript, is designed to target the EVM. Many, if not all, of these warnings we categorize are currently not integrated into any Solidity static analysis tools or the Solidity compiler itself. We focus the research on concerns discussed in the developer forums. Table 1 presents new categories of concerns which can be incorporated into a static analysis tool to improve smart contracts prior to deployment.

Table 1. New categories of concerns for contract static analysis

#	Type	Description
1	<i>Legal</i>	This can further be broken into different specific legal regulations. For example, the GDPR requires that data storage adheres to the right to be forgotten. As such, using methods like <i>selfdestruct</i> is not the same as deleting data from a hard disk. Depending on what data is stored on the public blockchain, it may or may not conform to legal requirements

(continued)

Table 1. (continued)

#	Type	Description
2	<i>Normalization prior to Input Validation</i>	Tools such as Dingman et al. [1], Feist et al. [28] and Slither [29] do consider certain input validation concerns such as integer overflow/underflow. Our research notes that input Unicode text can be encoded as a different byte array. Thus, they may either be missing proper normalization prior to filtering or holding an entirely unexpected values. Furthermore, any interaction with another contract needs to incorporate proper input validation
3	<i>Overflow/ Underflow</i>	Tools such as Dingman et al. [1] do consider overflow/underflow. Our research highlights specifically that different math operations (e.g. shifts, exponentiation, addition, subtraction, multiplication and division) need to each be considered during a static analysis. In addition, a safe math library could be employed
4	<i>Fixed point numbers</i>	Fixed point numbers are not yet fully supported by Solidity. They can be declared, but cannot be assigned. As such, smart contracts which integrate fixed point numbers need additional logic checks for contract correctness
5	<i>Unsafe Method Call</i>	Avoid using <code>.call()</code> whenever possible when executing another contract function. The <code>.call()</code> function bypasses type checking, function existence check, and argument packing
6	<i>Explicit Hardcoded Gas Values</i>	Although possible, best practice advises against specifying gas values explicitly (i.e. hardcoded), since the gas costs of opcodes can change at any future point
7	<i>Leaking Gas</i>	In certain circumstances, gas could be lost using <code>feed.info{value: x, gas: y}</code>

(continued)

Table 1. (continued)

#	Type	Description
8	<i>Judge Concerns</i>	This applies mainly when a contract acts as a judge for off-chain interactions (i.e. during disputes). A contract can be re-created at the same address after having been destroyed. Check the peculiarities (e.g. salt) so that the contract behaves as expected
9	<i>Unexpected Results from Multiple Assignments</i>	Assigning multiple variables when reference types are involved can lead to unexpected copying behavior
10	<i>Variable Sanitization</i>	Accessing variables of a type that spans less than 256 bits (for example uint64, address, bytes16 or byte), should not make any assumptions about bits not part of the encoding of the type. (i.e. do not assume all zero)

5 Solidity Source Code Analysis

In this section, we connect example suggested static analyses given in Section IV with Solidity v0.6.2 source code.

Figure 1 shows example contract code from Table 1 Category 1 calling the *selfdestruct* method. Depending on what information may be stored on the blockchain, certain legal concerns may not be satisfied. In addition, if any Ether is sent to a removed contract, the Ether is simply lost forever.

```
pragma solidity >0.6.1 <0.7.0;
function close() public onlyOwner {
    selfdestruct(owner);
}
```

Fig. 1. Static analysis category #1

Figure 2 shows example contract code of Category 3 given in Table 1 where even if a number is negative, it cannot be assumed by the developer that its negation will be positive.

```
int x = -2**255;
```

Fig. 2. Static analysis category #3

Figure 3 shows example contract code for Category 4 given in Table 1. Fixed point numbers can be declared in Solidity, but they are not completely supported. A static analysis could uncover usage or potential logic concerns, which need to be addressed.

```
pragma solidity >0.6.1 <0.7.0;
...
fixed x;
ufixed y;
ufixed128x18;
```

Fig. 3. Static analysis category #4

Figure 4 shows example contract code for the concern listed in Table 1 Category 7. Any Ether (i.e. smallest demonization Wei) sent to the contract is added to the total balance of that contract. Hardcoding gas values explicitly is not good practice, since the gas costs of opcodes can change in the future.

```
pragma solidity >=0.4.0 <0.7.0;
contract InfoFeed {
    function info() public payable returns (uint ret) { return 84; }
}
contract Consumer {
    InfoFeed feed;
    function callFeed() public { feed.info{value: 2, gas: 9}(); }}
```

Fig. 4. Static analysis category #7

6 Future Work

There is much future work in the secure software development of smart contracts. First, other blockchain languages could be considered for some of the categories of concerns we described in this paper such as legal. Second, some of the categories, such as legal, can further be broken into many different subtypes depending on the laws. For example, blockchains in healthcare settings in the United States of America needs to conform to HIPAA regulations. In addition, financial blockchains which interact with credit card data will need to conform to PCI compliance. Since smart contracts are immutable, once they are deployed correcting issues becomes very expensive to fix. Second, we can work to connect the issues we categorize in this paper with the Bug Framework as in Dingman et al. [1]. Third, we specifically examined Solidity v0.6.2. We could further examine concerns with other version of Solidity. Last, we could either develop a new static analysis tool or build onto an existing one to capture the findings we categorized in this paper.

7 Conclusions

Our research shows that there is still much work that remains to be done in the development of static analysis for secure software development of blockchain smart contracts. Solidity is a smart contract language designed to target the EVM. We identified new categories of development issues specific to Solidity v0.6.2. As smart contracts are immutable after deployment, they are not designed for patching software issues and software security concerns. Detecting smart contract coding concerns early in the secure software development lifecycle (sSDLC) is essential.

References

1. Dingman, W., et al.: Classification of smart contract bugs using the NIST bugs framework. In: 17th IEEE/ACIS International Conference on Software Engineering, Management and Applications (SERA 2019)
2. Xu, X., Pautasso, C., Zhu, L., Lu, Q., Weber, I.: A pattern collection for blockchain-based applications. In: Proceedings of the 23rd European Conference on Pattern Languages of Programs (EuroPLoP 2018). ACM, New York, NY, USA, Article 3, pp. 1–20 (2018). <https://doi.org/10.1145/3282308.3282312>
3. Mense, A., Flatscher, M.: Security vulnerabilities in Ethereum smart contracts. In: Proceedings of the 20th International Conference on Information Integration and Web-based Applications & Services (iiWAS2018), Indrawan-Santiago, M., Pardede, E., Salvadori, I.L., Steinbauer, M., Khalil, I., Anderst-Kotsis, G. (eds.). ACM, New York, NY, USA, pp. 375–380 (2018). <https://doi.org/10.1145/3282373.3282419>
4. Parker, T.: Smart Contracts: The Ultimate Guide to Blockchain Smart Contracts - Learn how to Use Smart Contracts for Cryptocurrency Exchange! CreateSpace Independent Publishing Platform, USA (2016)
5. Juels, A., Kosba, A., Shi, E.: The ring of gyges: investigating the future of criminal smart contracts. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). ACM, New York, NY, USA, pp. 283–295 (2016). <https://doi.org/10.1145/2976749.2978362>
6. Parker, T.: Smart Contracts: The Complete Step-By-Step Guide to Smart Contracts for Cryptocurrency Exchange. CreateSpace Independent Publishing Platform, USA (2017)
7. van der Laan, B., Ersoy, O., Erkin, Z.: MUSCLE: authenticated external data retrieval from multiple sources for smart contracts. In: Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing (SAC 2019). ACM, New York, NY, USA, pp. 382–391 (2019). <https://doi.org/10.1145/3297280.3297320>
8. Governatori, G., Idelberger, F., Milosevic, Z., Riveret, R., Sartor, G., Xu, X.: On legal contracts, imperative and declarative smart contracts, and blockchain systems. Artif. Intell. Law **26**(4), 377–409 (2018). <https://doi.org/10.1007/s10506-018-9223-3>
9. Lee, S., Park, S., Park, Y.B.: Formal specification technique in smart contract verification. In: 2019 International Conference on Platform Technology and Service (PlatCon), Jeju, Korea (South), 2019, pp. 1–4. <https://doi.org/10.1109/PlatCon.2019.8669419>
10. Hao, X., Xiao-Hong, S., Dian, Y.: Multi-agent system for e-commerce security transaction with block chain technology. In: 2018 International Symposium in Sensing and Instrumentation in IoT Era (ISSI), Shanghai, 2018, pp. 1–6 (2018). <https://doi.org/10.1109/ISSI.2018.8538253>

11. Naidu, V., Mudliar, K., Naik, A., Bhavathankar, P.P.: A fully observable supply chain management system using block chain and IOT. In: 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, 2018, pp. 1–4 (2018). <https://doi.org/10.1109/I2CT.2018.8529725>
12. Tang, H., Tong, N., Ouyang, J.: Medical images sharing system based on Blockchain and smart contract of credit scores. In: 2018 1st IEEE International Conference on Hot Information-Centric Networking (HotICN), Shenzhen, 2018, pp. 240–241 (2018)
13. Cheng, J., Lee, N., Chi, C., Chen, Y.: Blockchain and smart contract for digital certificate. In: 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, 2018, pp. 1046–1051 (2018). <https://doi.org/10.1109/ICASI.2018.8394455>
14. Afanasev, M.Y., Fedosov, Y.V., Krylova, A.A., Shorokhov, S.A.: An application of blockchain and smart contracts for machine-to-machine communications in cyber-physical production systems. In: 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, 2018, pp. 13–19 (2018). <https://doi.org/10.1109/ICPHYS.2018.8387630>
15. Bartoletti, M., Zunino, R.: BitML: a calculus for bitcoin smart contracts. In: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS 2018). ACM, New York, NY, USA, pp. 83–100 (2018). <https://doi.org/10.1145/3243734.3243795>
16. Schrans, F., Eisenbach, S., Drossopoulou, S.: Writing safe smart contracts in flint. In: Conference Companion of the 2nd International Conference on Art, Science, and Engineering of Programming (Programming 2018 Companion). ACM, New York, NY, USA, pp. 218–219 (2018). <https://doi.org/10.1145/3191697.3213790>
17. Neisse, R., Steri, G., Nai-Fovino, I.: A blockchain-based approach for data accountability and provenance tracking. In: Proceedings of the 12th International Conference on Availability, Reliability and Security (ARES 2017). ACM, New York, NY, USA, Article 14, p. 10 (2017). <https://doi.org/10.1145/3098954.3098958>
18. Kang, E.S., Pee, S.J., Song, J.G., Jang, J.W.: Blockchain based smart energy trading platform using smart contract. In: 2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIIC), Okinawa, Japan, 2019, pp. 322–325 (2019). <https://doi.org/10.1109/ICAIIIC.2019.8668978>
19. Bojanova, I., Black, P.E., Yesha, Y., Wu, Y.: The Bugs Framework (BF): a structured approach to express bugs. In: 2016 IEEE International Conference on Software Quality, Reliability and Security (QRS), Vienna, 2016, pp. 175–182 (2016). <https://doi.org/10.1109/QRS.2016.29>
20. Christidis, K., Devetsikiotis, M.: Blockchains and smart contracts for the internet of things. IEEE Access **4**, 2292–2303 (2016). <https://doi.org/10.1109/ACCESS.2016.2566339>
21. Luu, L., Chu, D.H., Olickel, H., Saxena, P., Hobor, A.: Making smart contracts smarter. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS 2016). ACM, New York, NY, USA, pp. 254–269 (2016). <https://doi.org/10.1145/2976749.2978309>
22. Wohrer, M., Zdun, U.: Smart contracts: security patterns in the ethereum ecosystem and solidity. In: 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE), Campobasso, 2018, pp. 2–8 (2018). <https://doi.org/10.1109/IWBOSE.2018.8327565>
23. Tikhomirov, S., Voskresenskaya, E., Ivanitskiy, I., Takhaviev, R., Marchenko, E., Alexandrov, Y.: SmartCheck: static analysis of ethereum smart contracts. In: Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB 2018). ACM, New York, NY, USA, pp. 9–16 (2018). <https://doi.org/10.1145/3194113.3194115>
24. Henney, K.: Inside requirements (2017). <https://www.slideshare.net/Kevlin/inside-requirements>

25. Atzei, N., Bartoletti, M., Cimoli, T.: A survey of attacks on ethereum smart contracts sok. In: Proceedings of the 6th International Conference on Principles of Security and Trust – vol. 10204, New York, NY, USA: SpringerSpringer-Verlag New York, Inc., 2017, pp. 164–186, ISBN: 978 978-3-662-54454-9 (2017). <https://doi.org/10.1007/978978-366236625445455-68>
26. Dika, A.: Ethereum smart contracts: Security vulnerabilities and security tools tools. Master Thesis, Norwegian University of Science and Technology (2017). https://brage.bibsys.no/xmlui/bitstream/handle/11250/2479191/18400_FULLTEXT. Accessed 27 February 2018
27. Solidity: Introduction to Smart Contracts (2020). <https://solidity.readthedocs.io/en/v0.6.2/introduction-to-smart-contracts.html>
28. Feist, J., Grieco, G., Groce, A.: Slither: a static analysis framework for smart contracts. In: 2019 IEEE/ACM 2nd International Workshop on Emerging Trends in Software Engineering for Blockchain (WETSEB), Montreal, QC, Canada, 2019, pp. 8–15 (2019)
29. Slither: GitHub: Static Analyzer for Solidity (2020). <https://github.com/crytic/slither>
30. Lindman, J., et al.: “The uncertain promise of blockchain for government”. OECD Working Papers on Public Governance, No. 43, OECD Publishing, Paris (2020). <https://doi.org/10.1787/d031cd67-en>
31. Sayeed, S., Marco-Gisbert, H., Caira, T.: Smart contract: attacks and protections. IEEE Access **8**, 24416–24427 (2020). <https://doi.org/10.1109/ACCESS.2020.2970495>



Promize - Blockchain and Self Sovereign Identity Empowered Mobile ATM Platform

Eranga Bandara¹(✉), Xueping Liang², Peter Foytik¹, Sachin Shetty¹, Nalin Ranasinghe³, Kasun De Zoysa³, and Wee Keong Ng⁴

¹ Old Dominion University, Norfolk, VA, USA

{cmedawer,pfoytik,sshetty}@odu.edu

² University of North Carolina at Greensboro, Greensboro, NC, USA

x_liang@uncg.edu

³ University of Colombo School of Computing, Colombo, Sri Lanka

{dnr,kasun}@ucsc.cmb.ac.lk

⁴ School of Computer Science and Engineering, Nanyang Technological University,

Singapore, Singapore

awkng@ntu.edu.sg

Abstract. Banks provide interactive money withdrawal/payment facilities, such as ATM, debit and credit card systems. With these systems, customers could withdraw money and make payments without visiting a bank. However, traditional ATM, debit and credit card systems inherit several weaknesses such as limited ATM facilities in rural areas, the high initial cost of ATM deployment, potential security issues in ATM systems, high inter-bank transaction fees etc. Through this research, we propose a blockchain-based peer-to-peer money transfer system “Promize” to address these limitations. The Promize platform provides a blockchain-based, low cost, peer-to-peer money transfer system as an alternative for traditional ATM system and debit/credit card system. Promize provides a self-sovereign identity empowered mobile wallet for its end users. With this, users can withdraw money from registered banking authorities (e.g. shops, outlets etc.) or their friends without going to an ATM. Any user in the Promize platform can act as an ATM, which is introduced as a mobile ATM. The Promize platform provides blockchain-based low-cost inter-bank transaction processing, thereby reducing the high inter-bank transaction fee. The Promize platform guarantees data privacy, confidentiality, non-repudiation, integrity, authenticity and availability when conducting electronic transactions using the blockchain.

Keywords: Blockchain · Self-sovereign identity · Smart contract · Electronic payments · Cloud computing

1 Introduction

Automatic Teller Machine (ATM) and debit/credit card systems are popular electronic transaction methods provided by banks. With these systems, customers can do money withdrawals to cash, payments, access their bank account information, check balance, etc. without visiting a bank teller. It is a pay-now payment system which is primarily used for small and macro transactions [40]. Today ATM's are a common feature in most banks revealing that the ATM is a successful technology that humans have adapted to. Even though these systems are popular electronic transaction systems, they inherit weaknesses and loopholes. Below is a list of some of them.

1. The initial cost of setting up an ATM system is very high, requiring separate internet connections, security systems, surveillance camera systems, and buildings to host.
2. ATMs are targets for various frauds and attacks [30]. To cope with these scenarios banks need to invest more money, time and well-trained staff members.
3. ATMs are not evenly scattered in the country. Most of the ATMs are located in urban areas, due to this reason customers have to travel long distances to use ATM facilities.
4. High inter-bank ATM transaction fees exist. All inter-bank transactions go through a third party centralized clearing agent. So for each inter-bank ATM transaction, customers need to pay a high transaction fee. For instance, in Sri Lanka, it costs LKR 20 per transaction.
5. High ATM transaction fee. For each transaction, customers need to pay a relatively high transaction fee. For example, Sri Lankan banks charge LKR 5 per single ATM transaction.
6. Anyone can steal the card number or the CVV number of debit or credit cards and make electronic payments. Physical cards are not required to make electronic payments.
7. Debit and credit cards are easily clone-able. Attackers can clone the card of a user and use it.
8. In many developing countries, national ATM switches are not implemented. Therefore, inter-banking ATM facilities are not available.

With this research, we propose a blockchain and self-sovereign identity [36]-based low-cost peer-to-peer money transfer system, “Promize” to address the above-mentioned challenges in traditional ATM systems and Debit/Credit card systems. Promize can be used as an alternative for traditional ATM systems(inter-bank ATM systems) and debit/credit card systems. With Promize, users are given the ability to withdraw funds using registered authorities or even through their friends without having to go to an ATM. Any user in the Promize platform can act as an ATM, which is introduced as a mobile ATM. All the Promize transactions are done through the self-sovereign identity empowered Promize mobile wallet application with using QR codes. Users need to link their bank account to the Promize wallet application. The Promize platform automatically links the bank accounts using the core-banking APIs. Once the wallet

is linked with a bank account, the Promize wallet can be used as an alternative for traditional debit/credit cards. Users can purchase goods from shops and pay via the Promize wallet instead of using a debit or credit card. Promize does not provide a traditional crypto-currency application with its blockchain. Instead, it provides a low-cost inter-bank money transfer and payment system with a blockchain. The blockchain system in the Promize platform is used to link multiple banks in the network. The Promize platforms' low-cost peer-to-peer money transfer system guarantees, non-repudiation, confidentiality, integrity, authenticity and availability of the transactions [26] with the help of blockchain integration. We have deployed a live version of the Promize platform at the Merchant Bank of Sri Lanka (MBSL) [33]. MBSL's ATMs are located only in urban areas, with no coverage in the rural areas of the country. Due to this, they provide ATM facilities to their customers in rural areas through the Promize platform. The main contributions of "Promize" are as follows.

1. Blockchain-based low-cost peer-to-peer money transfer system "Promize", introduced to address the challenges in traditional ATM systems and Debit/Credit card systems.
2. Self-Sovereign identity empowered mobile wallet (for Android/IoS) has been introduced to facilitate peer to peer money transfer and payment transactions. The transactions can be done with QR code scanning.
3. Promize platform based ATM system is introduced to MBSL bank customers in Sri Lanka to facilitate their day-to-day money withdrawal requirements.
4. Blockchain-based low-cost inter-banking transaction system has been introduced to Sri Lankan banks.

The rest of the paper is organized as below. Section 2 discusses the architecture of the Promize platform. Section 3, the functionality of the Promize platform. Section 4 describes the production deployment of the Promize platform in the MBSL bank, Sect. 5 discusses the evaluation of the introduced platform. Section 6 surveys related work. Section 7 concludes the Promize platform with suggestions for future work.

2 Promize Platform Architecture

2.1 Overview

Promize is a blockchain-based peer-to-peer money transfer platform. It can be used as an alternative for traditional ATM systems as well as traditional electronic payments. The architecture of the Promize platform is described in Fig. 1. It contains four main components.

1. Core bank layer - Each bank in the Promize platform has its own core banking system. The core bank maintains user bank accounts. It provides API's for account management, fund transfers etc.
2. Distributed ledger - All users' decentralized identities (DID) and transaction records are stored here.

3. Money transfer layer - Electronic transactions and verification happen here.
4. Communication layer - Data exchanges between Promize wallets happen in this peer-to-peer communication layer.

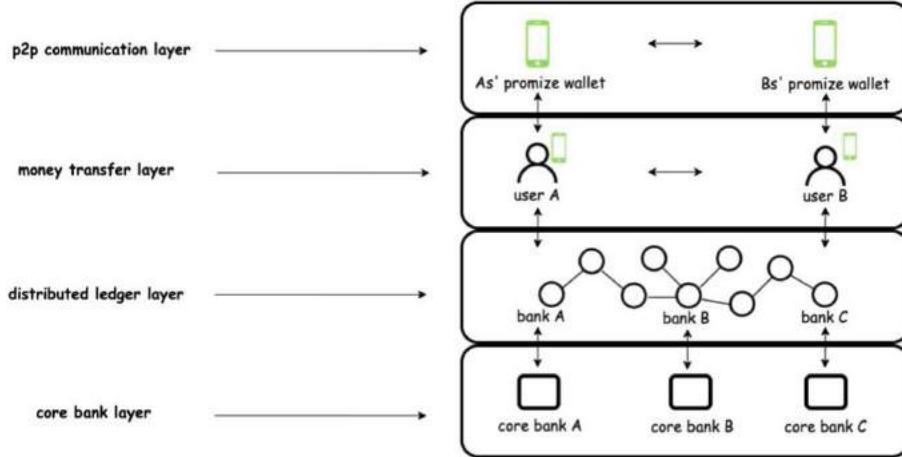


Fig. 1. Promize platform architecture. The Core Bank API maintains user bank accounts and supports fund transfers, etc. A distributed ledger is used to store user identities and transactions. Electronic money transfer transactions occur via the money transfer layer. The peer-to-peer communication layer is used to exchange transaction information.

2.2 Core Bank Layer

Each bank in the Promize platform has its own core banking system. The core bank maintains user bank account information, account balances, etc. It provides an API to search for user accounts and make fund transfers. The blockchain nodes deployed in the banks interact with their respective core banking APIs to search for user accounts, validate user accounts and perform fund transfers between accounts (for example, the blockchain node deployed in Bank A interacts with Bank A's core banking API). There are some common core banking systems which most banks use (e.g. Finacle core banking system [31]). These core banking systems expose JSON-based REST APIs or SOAP web service-based APIs to facilitate banking functions such as account creation, account validations, fund transfers, etc. The blockchain smart contracts in the Promize platform interact with these core banking APIs to perform user account validations and transfer funds between user accounts.

2.3 Distributed Ledger

The distributed ledger is the blockchain-based peer-to-peer storage system in the Promize platform. The blockchain can be deployed among multiple banks. Each bank in Promize can run their own blockchain node. These blockchain nodes are connected as a ring cluster, Fig. 2. Customers of each bank connect to their respective blockchain peer in the network. Blockchain stores all Promize platform users' digital identity (which is identified as DID or decentralized identity proof [6, 36]) information as self-sovereign identity. It also stores all the Promize transaction information in the blockchain. The user identity and transaction information which is stored in one banks' blockchain node will be synced with all other bank blockchain nodes by using the underlying blockchain consensus algorithm. The smart contracts running on the blockchain interact with the core bank API to facilitate customer bank account validations and fund transfers.

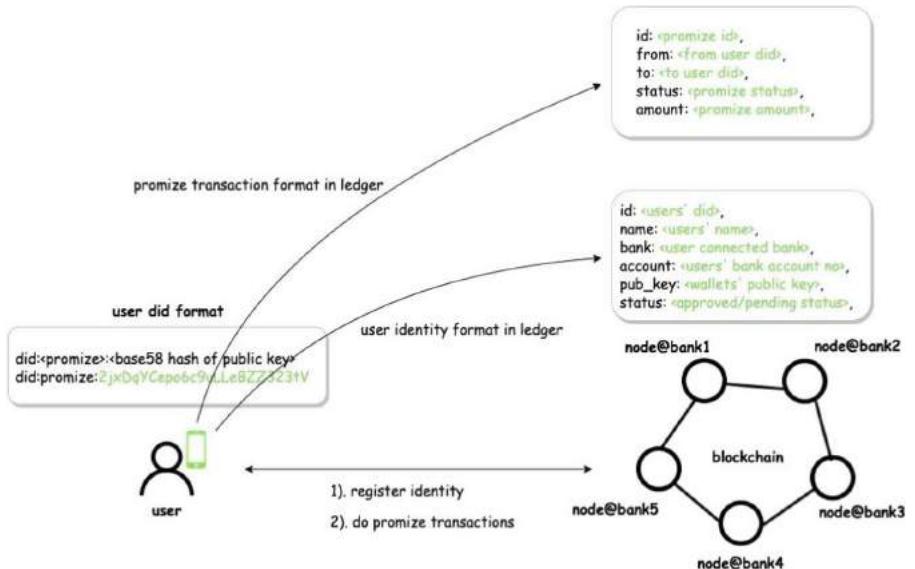


Fig. 2. Promize platform distributed ledger stores user identities and promize transactions. Promize mobile wallet interacts with the blockchain to create user identities and do promize transactions.

We have used Rahasak blockchain [7]; a highly scalable blockchain targeted for big data as the distributed ledger of the Promize platform. Rahasak comes with real-time transaction enabled “Validate-Execute-Group” blockchain architecture [5, 8] with using Paxos [29] and ZAB [25] based consensus. It introduced concurrent transaction enabled functional programming [22] and actor [20]-based “Apos” smart contract platform to facilitate blockchain functions [8]. All blockchain functions of the Promize platform are implemented with Apos smart contracts. “Identity smart contracts” are used to handle user identities and the “Promize smart contracts” are used to handle the Promize money transfer

transaction. With Rahasak blockchain we were able to support real-time transactions [16, 34], high transaction throughput, high scalability, backpressure operation handling features on Promize platform.

2.4 Money Transfer Layer

All peer-to-peer money transfers take place in the “Money transfer layer”. The users of the Promize platform will be given the Promize mobile wallet application. Peer-to-peer money transfers occur via this mobile wallet. Users first need to register on the Promize platform and link their bank account to the Promize mobile wallet. To use Promize, users need to have a bank account in any bank which participates in the Promize platform. When registering, the app will generate a public and private key pair which corresponds with the user/mobile wallet. The private key will be saved on the Keystore of the mobile application. The public key and base58 [17] hash of the public key will be uploaded to the blockchain along with other user attributes (e.g. account number, bank name etc.). The base58 hash of the public key will be used as the digital identity (DID) [36] of the user on the Promize platform. Figure 2 shows the format of the DID which is generated by the mobile wallet. This DID will be embedded to the QR code in the mobile app, which the user can show to any relevant parties (e.g. cashier, friend, bank branch officer etc.) when performing peer-to-peer money transfers via the Promize platform.

2.5 Peer to Peer Communication Layer

When making payments or performing transactions via the Promize platform, users need to exchange transaction details with the blockchain ledger and mobile wallet applications. The peer-to-peer communication layer used to exchange this transaction information between Promize mobile wallet applications and the blockchain. The peer-to-peer communication layer can be implemented with a TCP/WebSocket-based communication service or a push notifications service, similar to Firebase [27] (it works on top of 3G, 4G cellular network). In the Promize platform, we have used the Firebase push notification service to implement the peer-to-peer communication between mobile wallets.

3 Promize Platform Functionality

3.1 Use Case

In traditional scenarios, users need to go to a bank ATM or bank branch to withdraw money. With Promize, instead of going to a bank ATM, users can withdraw money from registered authorities of the Promize platform (e.g. shops, outlets, etc.) or a friend who has the Promize app installed. First, users register with the platform via the Promize mobile application. When a user registers, the mobile application captures user credentials, bank account information (Fig. 4) and sends a registration request to the “Identity smart contract” on the

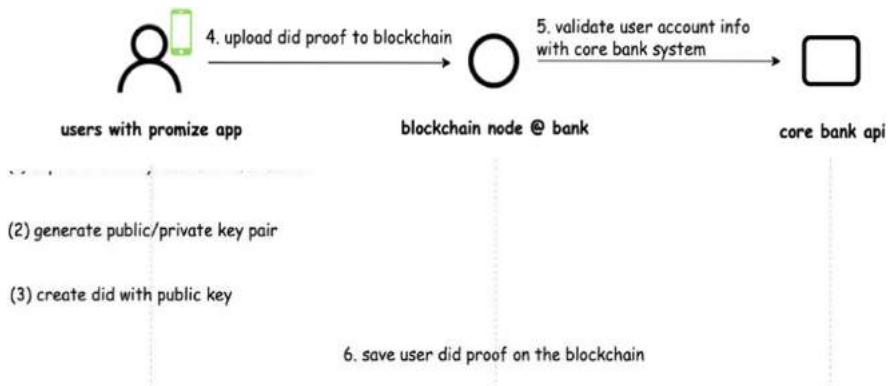


Fig. 3. User captures credentials and account information via the promize mobile wallet and register on the promize platform. The blockchain will store a proof of user credentials.

blockchain. The registration request contains `user id(DID)`, `user name`, `public key`, `user bank name` and `bank account number`. Upon receiving the registration request, smart contracts validate user credentials, bank account information and create an identity for the user in the blockchain, Fig. 3. The format of the user identity in the blockchain ledger can be seen in Fig. 2. When validating, it will connect to the corresponding core bank system at the bank and verify the user's bank account details (e.g. verify bank account number with identity number). For example, if a user belongs to Bank A, the blockchain node at Bank A will connect with its core bank system and validate the users' account information.

Assume two users (User A and User B) are registered on the Promize platform and link their bank accounts to the Promize mobile wallet. User A wants to withdraw LKR 1,000. User A goes to User B, who has already installed the Promize mobile app and registered on the Promize platform and asks for LKR 1000. If User B has LKR 1000 at hand, he could navigate to the “Send Promize” section on the mobile application and choose the amount to transfer. Now the app generates a QR code embedding User B's DID, account number and transfer amount, Fig. 4(c).

Now User A navigates to the “Received Promize” section in his mobile application and scans the QR code from User B's phone 5(a). Once the QR code is scanned, the application will navigate to the Promize receive confirmation screen Fig. 5(b), where User A can confirm it and submit a Promize transaction request to the blockchain, Fig. 5. When submitting the transaction, it sends a Promize create request to “Promize Smart Contract” on the blockchain, which checks the validity of the transaction(double-spend check) and user accounts. Promize create requests contain `transaction id`, `transaction type(create)`, `User A's did`, `User B's did` and `transaction amount`. If the transaction is valid, the blockchain will generate a corresponding random number (`transaction salt`)

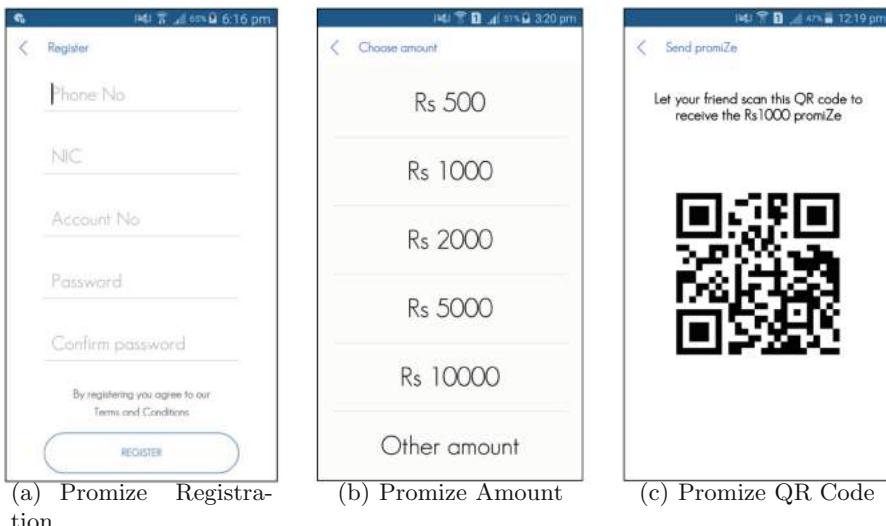


Fig. 4. Promize mobile wallet application registration. Promize sending user choose promize amount and initiate promize transaction. App generates a QR code embedding DID, account number and transfer amount.

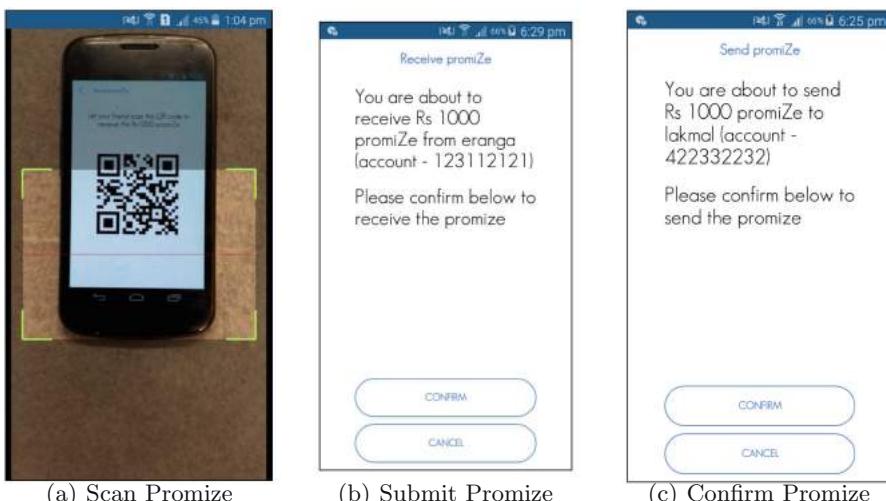


Fig. 5. Promize receiver scans the QR code and submits the transaction. A confirmation will be sent to promize sender via push notification. Once promize sender confirm it, the transaction will be approved.

and create a new Promize transaction in the blockchain ledger (status of the Promize transaction is marked as `pending`). The format of the Promize transaction in the blockchain ledger is shown in Fig. 2. Then it sends this transaction salt and other transaction details to User B's Promize mobile wallet as a notification message (e.g. Firebase push notification), Fig. 6. This notification message contains `transaction id`, `User A's did`, `User B's did`, `transaction amount` and `transaction salt`.

Once the notification message is received by User B's Promize application, it will navigate to the Promize send confirmation screen, Fig. 5. Then User B confirms it, thereby approving the Promize transaction. At confirmation, a Promize approval request will be sent with the received random number (transaction salt) to the “Promize Smart Contract” on the blockchain. The Promize approve request contains `transaction id`, `transaction type(approve)`, `User A's did`, `User B's did`, `transaction amount` and `transaction salt`. Then the smart contract will check the validity of the transaction parameters (transaction salt and accounts). When validating, it will compare the transaction salt and account numbers of the Promize transaction saved in the blockchain ledger. If the transaction parameters are valid, it will transfer the requested amount (LKR 1000) from User A's bank account to User B's bank account by communicating with the core banking service. If this transfer is successful, it will update the Promize transaction status to `approved`. Finally, the transaction status will be sent to both users' Promize mobile applications. Then User B physically gives LKR 1,000 to User A, Fig. 6.

The idea here is that User A physically receives money from User B which the bank electronically transfers to each other's accounts via Promize transactions. The same function can be used to purchase goods from shops. At the end of the purchase, users can send money to the shop's bank account via Promize transactions. Once the money is sent, the user receives goods.

3.2 Data Privacy and Security

The Promize platform guarantees data privacy, confidentiality, integrity, non-repudiation, authenticity and availability security features. To guarantee privacy, the Promize platform stores users' digital identity in the blockchain as a self-sovereign identity proof. Actual identity data such as ID numbers and signatures are stored on the users' physical mobile phone secure storage. When this information is needed for verification [19, 37], it will be directly fetched via the user's mobile application through push notifications. By using an SSI based approach, Promize platform addresses the common issues in centralized cloud-based storage platforms (e.g. lack of data immutability, lack of traceability). To guarantee confidentiality and integrity we have used RSA cryptography-based digital signature mechanism [24]. All data in the Promize platform are digitally signed by a corresponding party. The random number exchange with the QR code when performing a Promize transaction makes certain users meet in-person to carry out the transaction, which guarantees the non-repudiation. We have used a JWT based auth service to handle the authentication/authorization of the Promize

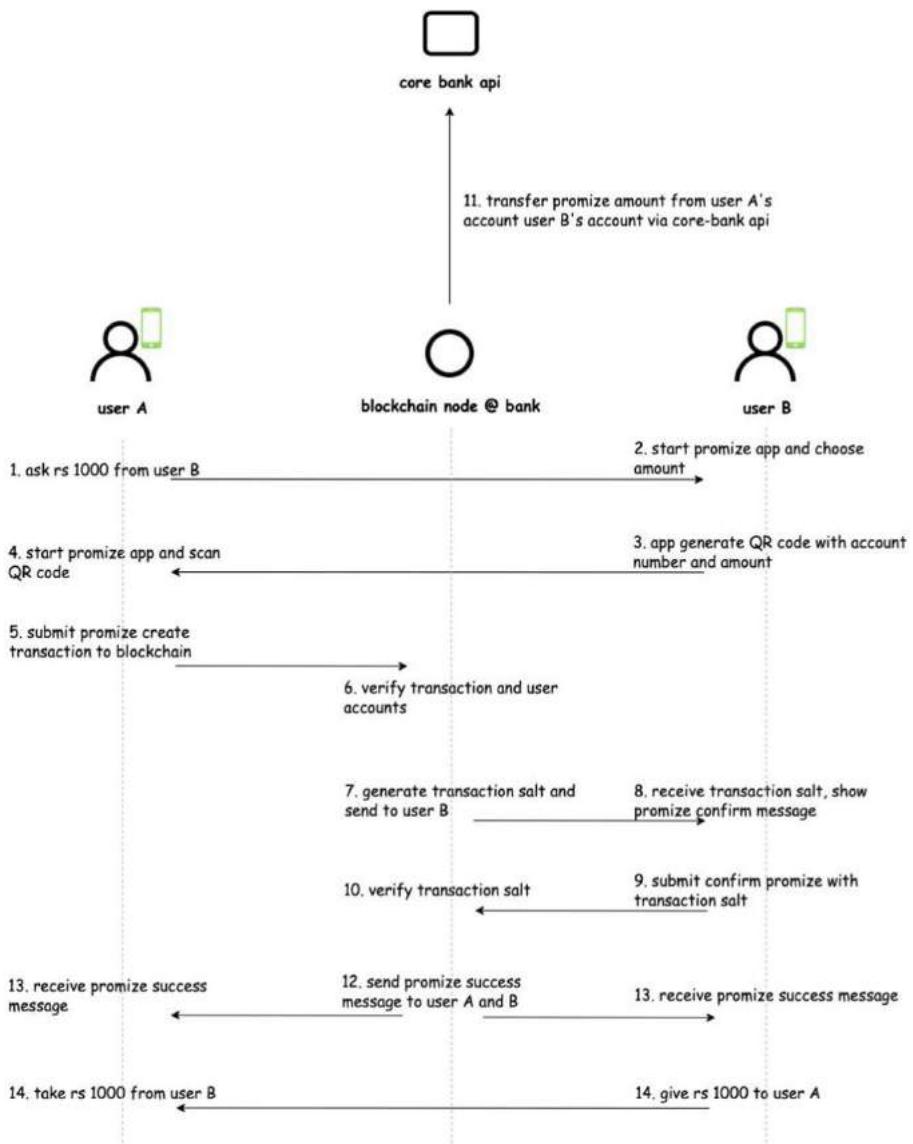


Fig. 6. Promize transaction flow between user A and user B. QR code is used to exchange information between mobile wallets.

platform. The users' authentication information (username/password fields) of Promize platform are stored in the JWT auth service [23]. Upon login, the user needs to send an authentication request to the auth service. Then, it verifies the credentials and returns the JWT auth token to the user. This auth token contains user permission, token validity time, digital signature etc. All subse-

quent requests need to be added to the JWT token and into the HTTP request header to perform the authentication and authorization. The token verification is handled by the gateway service in the Promize platform, Fig. 7.

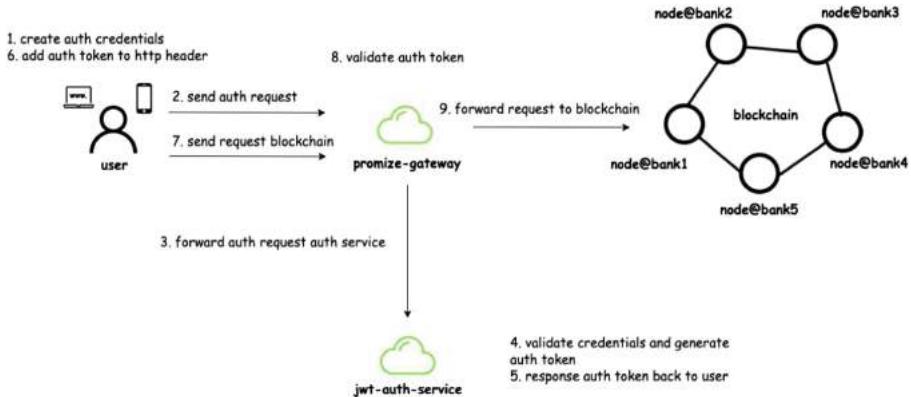


Fig. 7. JSON Web Token (JWT)-Based Auth service in the promize platform. Auth tokens will be issued by Auth service. Token validation happens at gateway service.

3.3 Low Cost Transactions

With Promize, the cost of electronic transactions can be easily reduced. Since multiple banks can share transaction and account information through the blockchain, it could facilitate low inter-bank transaction costs. Additionally, an inter-bank communication switch or an ATM switch is not required. With Promize, any user can act as an ATM, if they have money they can give it to their friends. If required, banks can appoint official Promize agents to act as ATMs. For example, a bank could appoint a trusted shop in a village as a Promize agent. Users in that village could go to this shop to withdraw money. They could also buy goods and pay via Promize transactions. It would reduce banks' costs to deploy and maintain physical ATMs and the operation costs of credit and debit cards. It's capable of providing services for a wide range of customers in rural areas, securely. In normal ATMs, customers take the money at the bank, which means the bank's money goes out. In Promize, users exchange money between accounts. For example, in above-mentioned scenario User B gives his physical money to User A, and Promize transfers money from User A's bank account to User B's bank account. In this case, the actual money at the bank remains the same. The money does not go out like in ATMs, it is another main advantage for the banks.

4 Promize Platform Production Deployment

4.1 Overview

We have deployed a live/production version of the Promize platform at the MBSL. Their branch network and ATM systems are mainly located in urban areas. This particular bank experiences a lack of ATM systems in rural areas. They use the Promize platform to facilitate banking and financial services (e.g. money withdrawals, electronic payments etc.) to people in rural villages in Sri Lanka. Promize mobile wallets are used as ATMs. They plan on appointing official bank agents in rural areas with a Promize mobile wallet. For example, supermarket/shops in the village could be bank agents. The agent's Promize mobile wallet will be installed in low-cost Sunmi android tablet devices. There is a built-in Bluetooth printer embedded in the Sunmi android tablet. Customers could go to a shop and withdraw money based on Promize protocol discussed in Sect. 3. At the end of the transaction, the cashier of the shop prints a receipt and gives it to the customer using the receipt printer embedded in the Sunmi tablet. This provides greater transparency and validity to Proimze transactions.

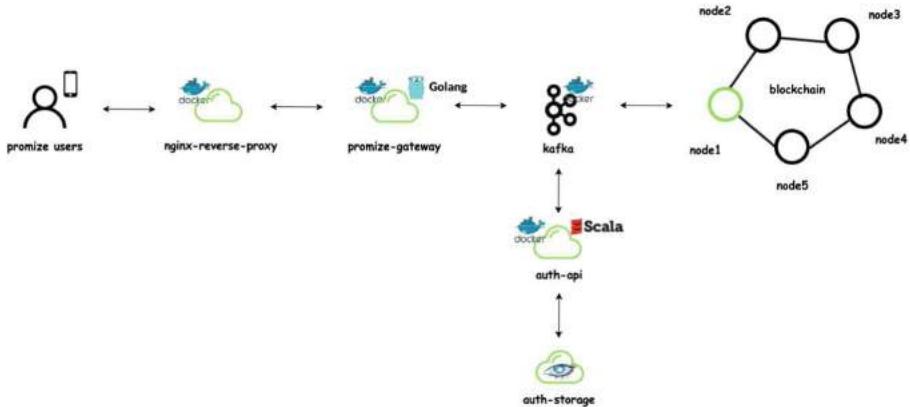


Fig. 8. Promize platform production system architecture which is deployed in MBSL Bank.

4.2 Service Architecture

We have built the Promize platform to support high scalability and high transaction load in the banking environment. To cope with high transaction load and back-pressure [14] operations, we have adopted reactive streams based approach with using Akka streams [2,13]. The Promize platform has been built using micro-services architecture [42]. All the services in the Promize platform are implemented as small services (micro-services) with the single responsibility principle. These services are dockerized [4,35] and deployed using the

Kubernetes [9] container orchestration system. Figure 8 shows the architecture of the Promize platform in MBSL. It contains the following services/components.

1. Nginx - Load balancer and reverse proxy service
2. Gateway service - Micro-services API gateway
3. Auth service - JWT based auth service
4. Apache Kafka - Micro-services message broker
5. Rahasak Blockchain - Blockchain ledger in the Promize platform

Nginx load balancer [38] is used as the reverse proxy in the Promize platform. The SSL certificates are set up in the Nginx and expose port 443 to the client. All mobile wallet applications connect with port 443 which is publicly exposed via public IP. The requests coming to Nginx will be redirected to the Gateway service, which is the microservices API gateway of the Promize platform. The gateway service is implemented with golang [3,39]. This service will validate auth tokens in client requests (HTTP header) and redirect them to the blockchain to handle Promize functionalities. Auth services are the JWT [23]-based auth service in the Promize platform. All client credentials(username, passwords) will be stored in auth services' auth storage. The auth service validates client credentials and issues auth tokens to clients. Each bank in the network has its own Gateway, Auth Service and Nginx-proxy. The bank clients(with Promize mobile applications) connect to the blockchain network via their respective banks' Nginx-proxy. Apache Kafka is used as the microservices message broker of the Promize platform. Inter-service communication takes place via Kafka message broker using JSON serialized messages. Every service in the Promize platform has its own Kafka topics to receive messages. We have used the highly scalable Rahasak blockchain to implement the Promize platform at MBSL. All blockchain functions are written with the Scala functional programming and Akka actor-based Aplos smart contract in the Rahasak blockchain. MBSL runs AS400-based core banking system [15]. It maintains customer accounts, account balance etc. There is a core banking API which is exposed to perform account inquiries, fund transfers etc. The blockchain smart contracts of the Promize platform interact with these core banking API perform user account validations and fund transfer between accounts.

5 Performance Evaluation

Performance evaluation of Promize was completed and is discussed. To obtain these results, we deployed the Promize platform with a multi-peer Rahasak blockchain cluster in AWS 2× large instances (16 GB RAM and 8 CPUs). Rahasak blockchain runs with 4 Kafka nodes, 3 Zookeeper nodes and Apache Cassandra [28] as the state database. The smart contracts on the Rahasak blockchain are implemented with Scala functional programming and the Akka actor-based [1,18] Aplos [8] smart contract platform. The evaluation results are obtained for the following, with a varying number of blockchain peers (1 to 5 peers) used in different evaluations.

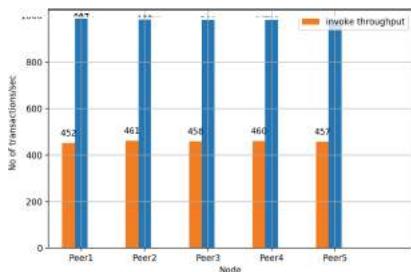


Fig. 9. Transaction throughput in the promize blockchain

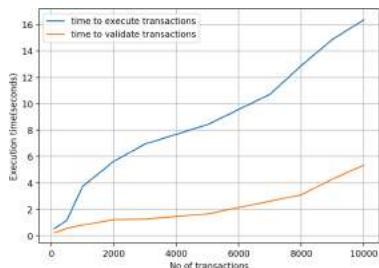


Fig. 10. Time to execute transactions and validate transactions in the promize platform.

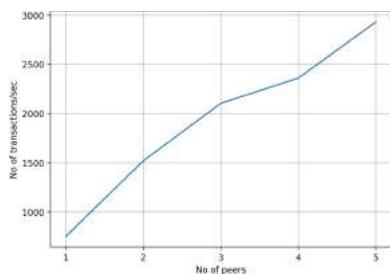


Fig. 11. Transaction scalability in the promize blockchain.

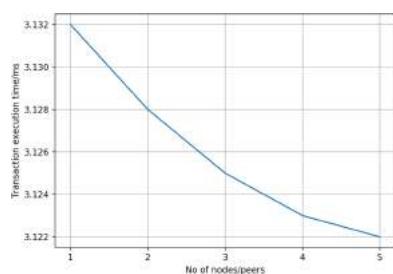


Fig. 12. Transaction latency in the promize blockchain.

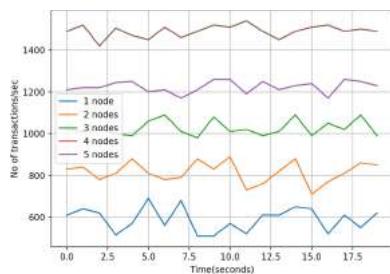


Fig. 13. Transaction execution rate with no blockchain peers in the promize blockchain.

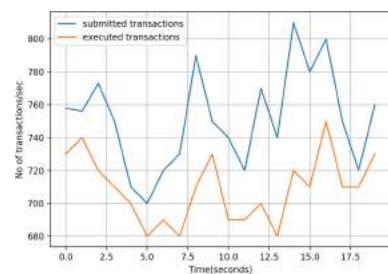


Fig. 14. Transaction execution rate and transaction submission rate in a single blockchain peer.

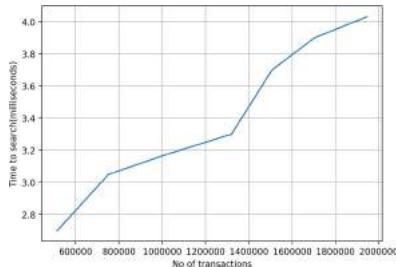


Fig. 15. Search performance of the Promize platform.

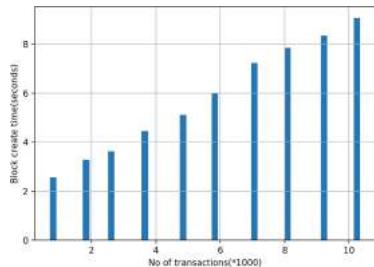


Fig. 16. Block creation time against the number of transactions in the block.

1. Transaction throughput of the blockchain
2. Transaction execution and validation time in Promize platform
3. Transaction scalability of the Promize platform
4. Transaction execution rate in the Promize platform
5. Search performance
6. Block creation time

5.1 Transaction Throughput

To complete this evaluation, we recorded the number of Promize create transactions and Promize query transactions that can be executed on each peer in the Promize platform. When creating Promize, an invoke transaction will be executed in the underlying blockchain. The invoke transaction would then create a record in the ledger and update the status of the assets in the blockchain. The query searches the status of the underlying blockchain ledger and can neither create transactions in the ledger nor update the asset status. We flooded concurrent transactions for each peer and recorded the number of completed results. As shown in Fig. 9 we have obtained consistent throughput on each peer of the Promize platform. Since query's do not update the ledger status, it has high throughput(2 times) compared to invoke transactions.

5.2 Transaction Execution and Validation Time

We evaluated the transaction execution and transaction validation time and recorded the time to execute and validate the different set of transactions (100, 500, 1000, 2000, 3000, 5000, 7000, 8000, 10000 transactions). The transaction validation time includes the double-spend checking time. Transaction execution time includes the double-spend checking time, ledger update time, data replication time. Figure 10 shows how the transaction execution time and validation time varies in different transaction sets.

5.3 Transaction Scalability

The number of transactions that can be executed (per second) over a number of peers in the network is recorded for this evaluation. We flooded concurrent transactions on each peer and recorded the number of executed transactions. Figure 11 shows transaction scalability results. Each node addition to the cluster has increased the transaction throughput in a nearly linear fashion. This means that the transaction latency will be decreased when adding blockchain peers to the cluster, Fig. 12. As peers are added, there is a diminishing return where the performance benefit will degrade if too many peers are added.

5.4 Transaction Execution Rate

Next, we evaluate the transaction execution rate in the Promize platform. We tested the number of submitted transactions and executed transactions in different blockchain peers, recording the time. Figure 13 shows how the transaction execution rate varies according to the different number of blockchain peers in the Promize platform. When the number of peers increases, the rate of executed transactions also increases, relatively. Figure 14 shows the number of executed transactions and submitted transactions in a single blockchain peer. There is a back pressure operation [14] between the rates of submitted transactions and executed transactions. We have used a reactive streaming-based approach with Apache Kafka to handle these backpressure operations in the Promize platform.

5.5 Search Performance

The Promize platform allows searching it's transaction and account information via the underlying Rahasak blockchain's Lucene Index-based search API. We evaluated this criteria by issuing concurrent search queries into Promize and computing the search time. As shown in Fig. 15, to search 2 million records connected it consumed 4 ms.

5.6 Block Generate Time

Finally, we have evaluated the time taken to create blocks in the underlying blockchain storage of the Promize platform. The statistics recorded against the number of transactions in a block. Block generation time depends on a). data replication time b). Merkel proof/block hash generate time c). transaction validation time. When the transaction count increases in the block, these factors will also increase. Due to this, when the transaction count increases, block generation time too increases correspondingly. As shown in Fig. 16 to increase a block when having 10k transaction, the platform consumes 8 s.

Table 1. Peer-to-peer electronic payment platform comparison

Platform	Architecture	Running blockchain	Scalability level	Payment type	SSI support	Privacy level
Promize	Decentralized	Rahasak	High	Bank	Yes	High
Mobile-ATM [26]	Centralized	N/A	Mid	Bank	No	Low
RSCoin [12]	Decentralized	RSCoin	Mid	Crypto	No	Mid
DTPS [21]	Decentralized	Ethereum	Low	Visa/MasterCard	No	Mid
BPCSS [11]	Decentralized	Bitcoin	Low	Crypto	No	Mid
BCPay [43]	Decentralized	Ethereum	Low	Crypto	No	Mid
TTPayment [32]	Decentralized	Bitcoin	Low	Crypto	No	Mid
Syscoin [41]	Decentralized	Bitcoin	Low	Crypto	No	Mid

6 Related Work

Much research has been conducted to improve the privacy/security features of electronic transactions. They seek to provide low-cost alternatives for traditional banking and inter-banking transactions [26]. In this section, we outline the main features and architecture of these research projects.

Mobile-ATM [26]. Mobile-ATM is a simple M-Commerce application that provides ATM services. The traditional ATM network is replaced with the proposed M-ATM system. This system is incorporated of a Bank, Customer and the M-ATM agent. Both, M-ATM agent and the customer in this system are required to have mobile phones set up to perform the functions of M-ATM. The bank has an M-ATM server that it hosts as a front-end interface and uses a back-end transaction management system. This tool proposes a new money withdrawal/deposit system (ATM system), enabling users to perform ATM transactions with their mobile phones with additional security features. The goal of this tool is to reduce barriers of using ATM systems while simultaneously improving security related to ATM transactions.

RSCoin [12], a centrally controlled cryptocurrency framework uses a sharding-based blockchain protocol that allows for better scalability and efficiency of the centrally-banked crypto-currency system. RSCoin utilizes a centralized monetary supply and distributed transaction ledger. A set of authorities are established and called `mintettes` that perform the transaction validation (double-spend checking). With a centralized monetary authority, RSCoin is able to scale better compared to other decentralized crypto-currencies. Currency supply is created and controlled by a central bank, making the cryptocurrency based on RSCoin significantly more palatable to governments. While monetary policy is centralized, RSCoin still provides strong transparency and auditability guarantees. Double spending is checked with a simple and fast mechanism and two-phase commit maintaining the integrity of the transaction ledger.

Delay-Tolerant Payment Scheme (DTPS) [21] proposes a cash-less payment system intended for rural villages where limited internet network connectivity exists. To work in an intermittent network environment, it uses blockchain

mining nodes located locally in villages that can handle the transaction processing and verification. Base stations are set up and run in these remote villages using technology such as Nokia Kuha connected to the public Internet via unreliable satellite links. This in turn offers 4g coverage to the village while providing connectivity to the wider internet intermittently. Banks control and initiate deployment of the system utilizing the intermittent connectivity and periodically monitor system operations of the villages. Ethereum [10] blockchain-based smart contracts are used for payment service management including, user account initiation, interactions with credit operator, and management of rewards for blockchain miners. VISA and MAsterCard handle all the local transaction verification through the Ethereum blockchain in which they have implemented payment gateways with.

BPCSS [11] proposes a Blockchain-based Payment Collection Supervision System(identified as BPCSS). Customers and merchandise stores can use this system to help manage transactions when they want to use the pervasive Bitcoin digital wallet. All transaction details are efficiently saved on a cloud database immediately after customers use their NFC-enabled Android smartphone App to purchase goods in the store using RFID-tags. Both customers and merchants are able to review transaction details in a sovereign way without the need of a traditional finance system. Results presented for the tool are preliminary using the Bitcoin Testnet but demonstrate the cost-effectiveness of the proposed tool to easily and effectively manage transactions.

BCPay [43] BCPay is introduced to provide secure and fair payment of outsourcing services in general without relying on third-party (trusted or non-trusted). Blockchain-based fair payment framework for outsourcing services on cloud and fog computing. BCPay can provide sound and robust payments without the need to have an organization facilitate it. This is achieved by an all-or-nothing checking-proof protocol that ensures the outsourcing service provider performs their service completely or pays a fee. Performance evaluations show that BCPay is able to process a high number of transactions in a short amount of time without a large computational cost.

Thing-to-Thing Payment (TTPayment) [32] is a bitcoin cryptocurrency-based automated micro payment platform for the Internet of Things. It allows devices to pay each other for services without a person required to interact. A proof-of-concept implementation exists of a smart cable that connects to a smart socket and pays for the electricity based on the draw without human interaction. Bitcoin is shown as a feasible payment solution for thing-to-thing payments, but high transaction fees exist when doing micro transactions. This is mitigated using the developed single-fee micro-payment protocol which aggregates multiple small transactions essentially batching them into one large transaction instead of many small transactions.

Syscoin [41] is a permissionless blockchain-based cryptocurrency providing blockchain-based e-commerce solutions for businesses of all sizes. Turing complete smart contracts built on the Bitcoin scripting system are used to control and interact with the system. Coin transactions are controlled by a hardened

layer of distributed consensus logic. Each smart contract (Syscoin service) uses this hardened layer while still retaining backwards compatibility with the Bitcoin protocol. Commercial developers want to use the most powerful network available (currently Bitcoin), Syscoin allows for the utilization of this network while providing security and efficiency.

The comparison summary of these platforms and the Promize platform is presented in Table 1. It compares the Architecture(Centralize/Decentralized), Running blockchain, Scalability level, Supported payment types(bank payments, Visa/Mastercard payments, cryptocurrency payments), SSI support, Privacy level details.

7 Conclusions and Future Work

With this research, we have proposed a blockchain-based low-cost alternative for traditional ATM systems and credit/debit card systems, “Promize”. With Promize, users can withdraw money from registered authorities or their friends without going to an ATM. Any user in the Promize platform can act as an ATM. All Promize transactions are done through the QR code-based Promize mobile wallet application. The Promize wallet application can be used as an alternative for traditional debit and credit cards. Users can purchase goods from shops and pay via the Promize wallet instead of using debit/credit cards. The Promize platform reduces the above-mentioned issues in ATM systems as well as debit/credit card systems. The Promize platform provides a secure, low-cost method of doing ATM, debit/credit card transactions while guaranteeing data privacy, confidentiality, integrity, non-repudiation, authenticity and availability security features.

We have proven the scalability and transaction throughput features of the Promize platform with empirical evaluations. Most recently we have deployed the 1.0 version of the Promize platform at the Merchant Bank of Sri Lanka.

Acknowledgments. This work was funded by the Department of Energy (DOE) Office of Fossil Energy (FE) (Federal Grant #DE-FE0031744).

References

1. Akka documentation
2. Akka streams documentation
3. The go programming language
4. Docker documentation, August 2018
5. Androulaki, E., et al. Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, p. 30. ACM (2018)
6. Baars, D.S.: Towards self-sovereign identity using blockchain technology. Master’s thesis, University of Twente (2016)
7. Bandara, E., et al.: Mystiko-blockchain meets big data. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 3024–3032. IEEE (2018)

8. Bandara, E., Ng, W.K., Ranasinghe, N.: Aplos: smart contracts made smart. In: BlockSys 2019 (2019)
9. Burns, B., Grant, B., Oppenheimer, D., Brewer, E., Wilkes, J.: Borg, omega, and kubernetes. Queue **14**(1), 70–93 (2016)
10. Buterin, V., et al.: A next-generation smart contract and decentralized application platform. white paper (2014)
11. Chen, P.-W., Jiang, B.-S., Wang, C.-H.: Blockchain-based payment collection supervision system using pervasive bitcoin digital wallet. In: 2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 139–146. IEEE (2017)
12. Danezis, G., Meiklejohn, S.: Centrally banked cryptocurrencies. *arXiv preprint arXiv:1505.06895* (2015)
13. Davis, A.L.: Akka streams. In: Reactive Streams in Java, pp. 57–70. Springer (2019)
14. Destounis, A., Paschos, G.S., Koutsopoulos, I.: Streaming big data meets back-pressure in distributed network computation. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, pp. 1–9. IEEE (2016)
15. Ebadi, Z.: Advance banking system features with emphasis on core banking. In: The 9th International Conference on Advanced Communication Technology, vol. 1, pp. 573–576. IEEE (2007)
16. Eykholt, E., Meredith, L.G. and Denman, J.: Rchain architecture documentation (2017)
17. Fisher, J., Sanchez, M.H.: Authentication and verification of digital data utilizing blockchain technology, September 29 2016. US Patent App. 15/083,238
18. Gupta, M.: Akka essentials. Packt Publishing Ltd. (2012)
19. Hammudoglu, J.S., et al.: Portable trust: biometric-based authentication and blockchain storage for self-sovereign identity systems. *arXiv preprint arXiv:1706.03744* (2017)
20. Hewitt, C.: Actor model of computation: scalable robust information systems. *arXiv preprint arXiv:1008.1459* (2010)
21. Yining, H., et al.: A delay-tolerant payment scheme based on the ethereum blockchain. IEEE Access **7**, 33159–33172 (2019)
22. Hughes, J.: Why functional programming matters. Comput. J. **32**(2), 98–107 (1989)
23. Jones, M.B.: The emerging json-based identity protocol suite. In: W3C Workshop on Identity in the Browser, pp. 1–3 (2011)
24. Jonsson, J., Kaliski, B.: Public-key cryptography standards (pkcs) # 1: Rsa cryptography specifications version 2.1. Technical report, RFC 3447, February 2003
25. Junqueira, F.P., Reed, B.C., Serafini, M.: Zab: high-performance broadcast for primary-backup systems. In: 2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), pp. 245–256. IEEE (2011)
26. Karunaratne, A., De Zoysa, K., Muftic, S.: Mobile ATM for developing countries, January 2008
27. Khawas, C., Shah, P.: Application of firebase in android app development-a study. Int. J. Comput. Appl. **179**(46), 49–53 (2018)
28. Lakshman, A., Malik, P.: Cassandra: a decentralized structured storage system. ACM SIGOPS Oper. Syst. Rev. **44**(2), 35–40 (2010)
29. Lamport, L.: The part-time parliament. ACM Trans. Comput. Syst. (TOCS) **16**(2), 133–169 (1998)

30. Li, Z., Sun, Q., Lian, Y., Giusto, D.D.: An association-based graphical password design resistant to shoulder-surfing attack. In: 2005 IEEE International Conference on Multimedia and Expo, pp. 245–248. IEEE (2005)
31. Luka, M.K. and Frank, I.A.: The impacts of ICTs on banks. Editorial Preface **3**(9), (2012)
32. Lundqvist, T., de Blanche, A., Andersson, H.R.H.: Thing-to-thing electricity micro payments using blockchain technology. In: 2017 Global Internet of Things Summit (GIoTS), pp. 1–6. IEEE (2017)
33. MBSL. MBSL bank
34. McConaghy, T., et al.: Bigchaindb: a scalable blockchain database. white paper, BigChainDB (2016)
35. Merkel, D.: Docker: lightweight Linux containers for consistent development and deployment. *Linux J.* **2014**(239), 2 (2014)
36. Mühle, A., Grüner, A., Gayvoronskaya, T., Meinel, C.: A survey on essential components of a self-sovereign identity. *Comput. Sci. Rev.* **30**, 80–86 (2018)
37. Othman, A., Callahan, J.: The horcrux protocol: a method for decentralized biometric-based self-sovereign identity. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE (2018)
38. Reese, W.: Nginx: the high-performance web server and reverse proxy. *Linux J.* **2008**(173), 2 (2008)
39. Schmager, F., Cameron, N., Noble, J.: Evaluating the go programming language with design patterns. In: Evaluation and Usability of Programming Languages and Tools, p. 10. ACM (2010)
40. Schwiderski-Grosche, S., Knospe, H.: Secure mobile commerce. *Electron. Commun. Eng. J.* **14**(5), 228–238 (2002)
41. Sidhu, J.: Syscoin: a peer-to-peer electronic cash system with blockchain-based services for e-business. In: 2017 26th International Conference on Computer Communication and Networks (ICCCN), pp. 1–6. IEEE (2017)
42. Thönes, J.: Microservices. *IEEE softw.* **32**(1), 116–116 (2015)
43. Zhang, Y., Deng, R.H., Liu, X., Zheng, D.: Blockchain based efficient and robust fair payment for outsourcing services in cloud computing. *Inf. Sci.* **462**, 262–277 (2018)



Investigating the Robustness and Generalizability of Deep Reinforcement Learning Based Optimal Trade Execution Systems

Siyu Lin^(✉) and Peter A. Beling

University of Virginia, Charlottesville, VA, USA
{sl15tb,pb3a}@virginia.edu

Abstract. Recently, researchers from both academia and the financial industry have attempted to leverage the cutting edge techniques of Deep Reinforcement Learning (DRL) to develop autonomous trade execution systems. They desire a system that could learn from the trade data and develop trade execution strategies to optimize the execution costs. While several researchers have reported success in developing such autonomous trade execution systems, none of them have investigated the robustness of the systems. Despite the powerfulness of DRL, the overfitting and generalization remain a challenge in DRL. For real-world applications, especially in financial trading, the DRL system's robustness is critical, as the cost of wrong decisions could be devastating. In our experiment, we investigate the robustness of the DRL based autonomous trade execution systems by applying the policies learned from one stock to other stocks directly without any refining or training. As different stocks have different dynamics, they might serve as suitable environments to investigate the systems' capability to generalize. The result suggests that the DRL systems have generalized well on other stocks without further refining or training on the specific stock's historical trade data. The result is significant from two perspectives: 1) It suggests that the DRL based trade execution systems can generalize well, which gives us and potential users more confidence in their performances; 2) It also presents an opportunity to develop a more sample efficient system.

Keywords: Deep reinforcement learning · Optimal trade execution · Robustness · Generalizability

1 Introduction

In the modern financial market, electronic trading has gradually replaced the traditional floor trading and is constituting a majority of the overall trading volumes. Nowadays, brokerage firms compete with each other intensively to provide better execution quality to retail or institutional investors. Optimal trade execution, which concerns how to minimize trade execution costs of trading a

certain amount of shares within a specified period, is a critical factor of execution quality.

From the regulatory perspective, brokers are legally required to execute orders on behalf of their clients to ensure the best execution possible. In the US, such practices are monitored by the Securities and Exchange Commission (SEC) and Financial Industry Regulatory Authority (FINRA). In Europe, MiFID II, a legislative framework instituted by the European Union, regulates financial markets and improves investor protection.

Encouraged by recent successful applications of Deep Reinforcement Learning in many challenging areas, researchers from academia and the financial industry have attempted to leverage the DRL techniques to develop autonomous trade execution systems, which could learn from the historical trade data and make optimized trade execution decisions autonomously. The successful development of such systems would significantly improve the efficiency of the existing manually designed, rule-based system or human execution trader and would dramatically change the brokerage industry.

Nevmyvaka, Feng, and Kearns have published the first large-scale empirical application of RL to optimal trade execution problems [10]. Hendricks and Wilcox propose to combine the Almgren and Chriss model (AC) and RL algorithm and to create a hybrid framework mapping the states to the proportion of the AC-suggested trading volumes [3]. To address the high dimensions and the complexity of the underlying dynamics of the financial market, Ning et al. [11] adapt and modify the Deep Q-Network (DQN) [9] for optimal trade execution, which combines the deep neural network and the Q-learning, and can address the curse of dimensionality challenge faced by Q-learning. Lin and Beling [6] analyze and demonstrate the flaws when applying a generic Q-learning algorithm and propose a modified DQN algorithm to address the zero-ending inventory constraint.

Although the researchers mentioned above have successfully developed autonomous trade execution systems, none of them have provided evidence of the systems' robustness, which might lead to concerns from potential users. In this article, we investigate the robustness of the DRL based autonomous trade execution systems by applying the learned policies from one stock to others without further training or refining. The result suggests that the performances of autonomous trade execution systems are robust, and it could learn essential underlying dynamics and can leverage that knowledge to different stocks.

2 DRL Based Optimal Trade Execution

In this section, we briefly review some DRL based optimal trade execution systems, including the states, reward function, assumptions, and discuss how to formulate optimal trade execution as a DRL problem. Deep Reinforcement Learning techniques could be divided into two main categories: 1) Value-based Q learning algorithm such as Deep Q-Network (DQN) proposed by Google DeepMind [9]; and 2) Policy Gradient method such as Proximal Policy Optimization (PPO)

proposed by OpenAI [12]. In the following sections, we describe two DRL based trade execution systems based on DQN and PPO, respectively.

2.1 Preliminaries

Deep Q-Network. Similar to Q-learning, the goal of the DQN agent is to maximize cumulative discounted rewards by making sequential decisions while interacting with the environment. The main difference is that DQN is using a deep neural network to approximate the Q function. At each time step, we would like to obtain the optimal Q function, which obeys the *Bellman equation*. The rationale behind *Bellman equation* is straightforward: if the optimal action-value function $Q^*(s', a')$ was completely known at next step, the optimal strategy at current step would be to maximize $E[r + \gamma Q^*(s', a')]$, where γ is the discounted factor [8].

$$Q^*(s, a) = E_{s' \sim \Phi}[r + \gamma \max_{a'} Q^*(s', a') | s, a] \quad (1)$$

The Q-learning algorithm estimates the action-value function by iteratively updating $Q_{i+1}(s, a) = E_{s' \sim \Phi}[r + \gamma \max_{a'} Q_i(s', a') | s, a]$. It has already been demonstrated that the action-value Q_i would eventually converge to the optimal action-value Q^* as $i \rightarrow \infty$ [13]. The Q-learning iteratively updates the Q table to obtain an optimal Q table. However, it suffers from the curse of dimensionality and is not scalable to large scale problems. The DQN algorithm has addressed this challenge faced by Q-learning. It trains a neural network model to approximate the optimal Q function by minimizing a sequence of loss function $L_i(\theta_i) = E_{s, a \sim \rho(\cdot)}[(y_i - Q(s, a; \theta_i))^2]$ iteratively, where $y_i = E_{s' \sim \Phi}[r + \gamma \max_{a'} Q^*(s', a'; \theta_{i-1}) | s, a]$ is the target function and $\rho(s, a)$ refers to the probability distribution of states s and actions a. In the DQN algorithm, the target function is usually freezed for a while when optimizing the loss function to prevent the instability caused by the frequently shift in target function. The gradient could be obtained by differentiating the loss function $\nabla_{\theta_i} L_i(\theta_i) = E_{s, a \sim \rho(\cdot); s' \sim \Phi}[(r + \gamma \max_{a'} Q^*(s', a'; \theta_{i-1}))_i - Q(s, a; \theta_i)] \nabla_{\theta_i} Q(s, a; \theta_i)$.

The model weights could be estimated by optimizing the loss function through stochastic gradient descent algorithms. In addition to the capability of handling high-dimensional problems, the DQN algorithm is also *model-free* and has no assumption about the dynamics of the environment. It learns about the optimal policy by exploring the state-action space.

The Q-learning algorithm is a *model-free* technique that has no assumptions on the environment. However, the curse of dimensionality has limited its application in high-dimensional problems. Deep Q-Network seems to be a natural choice because it can handle high-dimensional problems while inheriting Q-learning's ability to learn from its experiences. Given the abundant amount of information available in the limit order book, we believe that the Deep Q-Network could be more suitable to utilize those high dimensional market microstructure information. In this section, we provide the DQN formulation for the optimal trade execution problem and describe the state, action, reward, and the algorithm used in the experiment.

Proximal Policy Optimization. The PPO algorithm is a policy gradient algorithm proposed by OpenAI [12]. It becomes one of the most popular RL methods due to its state-of-the-art performance as well as its sample efficiency and easy implementation. To optimize policies, it alternates between sampling data and optimizing a “surrogate” objective function.

PPO is an on-policy algorithm and applies to both discrete and continuous action spaces. PPO-clip updates policies via

$$\theta_{k+1} = \arg \max_{\theta} E_{s,a \sim \pi_{\star_k}} [L(s, a, \theta_k, \theta)] \quad (2)$$

where π is the policy, θ is the policy parameter, k is the k^{th} step, a and s are action and state respectively. It typically takes multiple steps of SGD to optimize the objective L

$$\begin{aligned} L(s, a, \theta_k, \theta) = \\ \min \left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s, a), \text{clip}\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}, 1 - \epsilon, 1 + \epsilon\right) A^{\pi_{\theta_k}}(s, a) \right) \end{aligned} \quad (3)$$

where $A^{\pi_{\theta_k}}(s, a)$ is the advantage estimator. The clip term in Eq. 3 clips the probability ratio and prevents the new policy going far away from the old policy¹ [12].

2.2 Problem Formulation

In this article, we investigate two DRL based systems: 1) A modified DQN algorithm proposed by Lin and Beling [6]; 2) PPO with Long short-term memory (LSTM) algorithm proposed by Lin and Beling [7]. In the following sections, we describe the problem formulation for the two algorithms, respectively.

States. The state is a vector to describe the current status of the environment. In the settings of optimal trade execution, the states consist of 1) Public state: market microstructure variables including top 5 bid/ask prices and associated quantities, bid/ask spread, 2) Private state: remaining inventory and elapsed time, 3) Derived state: we also derive features based on historical LOB states to account for the temporal component in the environment². The derived features could be grouped into three categories: 1) Volatility in VWAP price; 2) Percentage of positive change in VWAP price; 3) Trends in VWAP price. More specifically, we derive the features based on the past 6, 12, and 24 steps (each step is a 5-second interval), respectively. Additionally, we use several trading volumes: 0.5*TWAP, 1*TWAP, and 1.5*TWAP³ to compute the VWAP price.

¹ <https://spinningup.openai.com/en/latest/algorithms/ppo.html>.

² For example, the differences of immediate rewards between the current time and arrival time at various volume levels, manually crafted indicators to flag specific market scenarios (i.e., regime shift, a significant trend in price changes, and so on).

³ TWAP represents the trading volume of the TWAP strategy in one step.

It is straightforward to derive features in 1). For 2), we record the steps that the current VWAP price increases compared with the previous step and compute the percentage of the positive changes. For 3), we calculate the difference between the current VWAP prices and the average VWAP prices in the past 6, 12, and 24 steps.

In the meanwhile, researchers have attempted to leverage LSTM networks to process raw level 2 microstructure data to account for time dependencies among the data, which could reduce the time and efforts spent on manually deriving attributes.

Actions. In this article, we choose different numbers of shares to trade based on the liquidity of the stock market. As the purpose of the research is to evaluate the capability of the proposed framework to balance the liquidity risk and timing risk, we choose the total number of shares to ensure that the TWAP orders⁴ can consume at least the 2nd best bid price on average. The total shares to trade for each stock are illustrated in Table 1. In the optimal trade execution framework, we set the range of actions from 0 to 2TWAP and set the minimal trading volume for each stock.

Table 1. # of shares to trade for each stock in the article

Ticker	Trading shares	Ticker	Trading shares
FB	6000	GS	300
GOOG	300	CRM	1200
NVDA	600	BA	300
MSCI	300	MCD	600
TSLA	300	PEP	1800
PYPL	2400	TWLO	600
QCOM	7200	WMT	3600

Rewards. The reward structure is the key to the DRL algorithm and should be carefully designed to reflect the DRL agent's goal. Otherwise, the DRL agent cannot learn the optimal policy as expected. In previous work [3, 10, 11], researchers use the IS⁵, a commonly used metric to measure execution gain/loss, as the immediate reward received after execution at each non-terminal step. There is a common misconception that the IS is a direct optimization goal. The IS compares the model performance to an idealized policy that assumes infinite liquidity

⁴ TWAP order = $\frac{\text{Total number of shares to trade}}{\text{Total \# of periods}}$.

⁵ Implementation Shortfall = arrival price \times traded volume – executed price \times traded volume.

at the arrival price. In the real world, brokers often use TWAP and VWAP as benchmarks. IS is usually used as the optimization goal because TWAP and VWAP prices could only be computed until the end of the trading horizon and cannot be used for real-time optimization. Essentially, IS is just a surrogate reward function, but not a direct optimization goal. Hence, a reward structure is good as long as it can improve model performance.

Shaped Rewards. In contrast to previous work [3, 10, 11], the DQN algorithm proposed by Lin and Beling [6] uses a shaped reward structure. Lin and Beling [6] point out that IS reward is noisy and nonstationary, which makes the learning process difficult. They proposed the shaped reward structure aiming to remove the impact of trends to standardize the reward signals.

Sparse Rewards. Although the shaped reward structure seems to generalize reasonably well on the rest equities as demonstrated in their article, the designing of a complicated reward structure is time-consuming and has a risk of overfitting in real-world applications. In contrast to previous approaches, Lin and Beling propose to use a sparse reward, which only gives the agent a reward signal based on its relative performance compared against the TWAP baseline model [7].

Zero Ending Inventory Constraint. In the real world business, the brokers receive contracts or directives from their clients to execute a certain amount of shares within a specific time. For the brokers, it is mandatory to liquidate all the shares by the end of the trading period. Lin and Beling [6] modify the Q-function update to combine the last two steps for Q-function estimation to incorporate the zero-ending inventory constraint. We leverage their methods by combining the last two steps.

2.3 Architecture

DQN Architecture and Extensions. The vanilla DQN algorithm has addressed the instability issues of using the nonlinear function approximation by techniques such as experience replay and the target network. In this article, we also incorporate several other techniques to further improve its stability and performance, as illustrated in Fig. 1. The architectures are chosen in Ray’s Tune platform by comparing model performances of all combinations of these extensions on Facebook data only, which is equivalent to ablation studies. A brief description of the DQN architecture and the techniques we use are below.

In the DQN algorithm, we use the exact same architecture as in Lin and Beling’s article: a fully connected feedforward neural network with two hidden layers, 128 hidden nodes in each hidden layer, and ReLU activation function in each hidden node. The input layer has 51 nodes, including private attributes such as remaining inventory and time elapsed as well as derived attributes based on public market information. The output has 21 nodes with a linear function as the activation function. The Adam optimizer is chosen for weights optimization [6].

It is well known that the maximization step tends to overestimate the Q function. van Hasselt [14] addresses this issue by decoupling the action selection

from its evaluation. In 2016, van Hasselt et al. [15] had successfully integrated this technique with DQN and had demonstrated that the double DQN could reduce the harmful overestimation and improve the performance of DQN. The only change is in the loss function below.

$$\begin{aligned} *_{\alpha_i} L_i(\gamma_i) = & E_{s,a \sim \alpha(\cdot); s' \sim \varepsilon} [((r + \gamma Q(s', \max_{a'} Q^*(s', a'; \gamma_{i-1})))_i \\ & - Q(s, a; \gamma_i)) *_{\alpha_i} Q(s, a; \gamma_i)] \end{aligned} \quad (4)$$

The dueling network architecture consists of two streams of computations with one stream representing state values and the other one representing action advantages. The two streams are combined by an aggregator to output an estimate of the Q function. Wang et al. have demonstrated the dueling architecture's capability to learn the state-value function more efficiently [16]. Noisy nets are proposed to address the limitations of the ϵ -greedy exploration policy in vanilla DQN. A linear layer with a deterministic and noisy stream, as demonstrated in Eq. 5, replaces the standard linear $y = b + Wx$. The noisy net enables the network to learn to ignore the noisy stream and enhance the effectiveness of exploration [2].

$$y = (b + Wx) + (b_{noisy} \odot \epsilon^b + (W_{noisy} \odot \epsilon^w)x) \quad (5)$$

PPO Architecture. Unlike the DQN algorithm, the PPO algorithm optimizes over the policy π_t directly and finds the optimal state-value function V_t^* . We leverage Ray's Tune platform to select the network architecture and hyperparameters by comparing model performances on Facebook data only, which is equivalent to ablation studies. The selected network architectures are illustrated in Fig. 1.

In the PPO algorithm, we have implemented the same network architecture as in Lin and Beling's article [7]: FCN with two hidden layers, 128 hidden nodes in each hidden layer, and ReLU activation function in each hidden node. The input layer has 22 nodes, including private attributes such as remaining inventory and time elapsed as well as the LOB attributes such as 5-level bid/ask prices and volumes. After that, we concatenate the model output and the previous reward and action and feed them to an LSTM network with a cell size of 128. The LSTM outputs the policy π_t and the state-value function V_t^* .

Unlike the feedforward neural networks, LSTM has feedback connections, which allows it to store information and identify temporal patterns. LSTM networks are composed of a number of cells, and an input gate, an output gate, and a forget gate within each cell regulate the flow of information into and out of the cell. It was proposed by Sepp Hochreiter and Jürgen Schmidhuber to solve the exploding and vanishing gradient problems encountered by the traditional recurrent neural network (RNN) [4]. Instead of manually designing attributes to represent the temporal patterns, we leverage LSTM to process the sequential LOB data and automatically identify temporal patterns within the data.

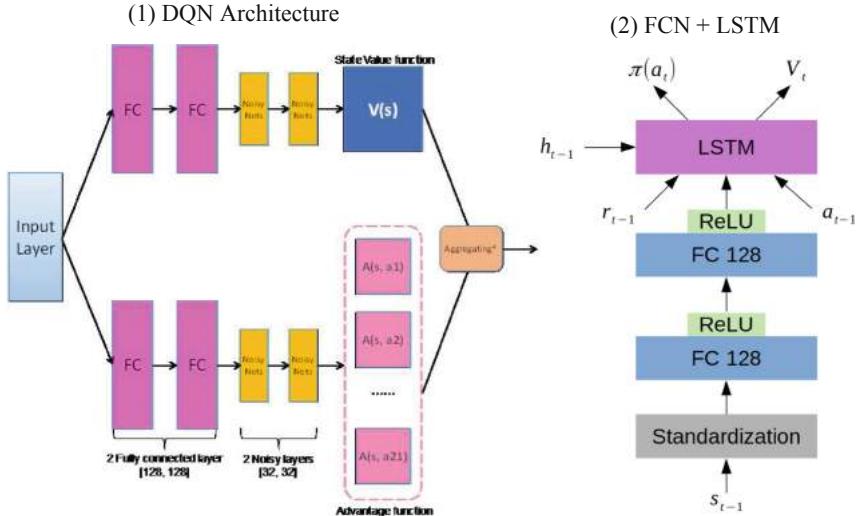


Fig. 1. DQN and PPO architectures.

2.4 Assumptions

The most critical assumption in our experiment is that the actions that the DRL agents have only a temporary market impact, and the market is resilient and will bounce back to the equilibrium level at the next time step. The market resilience assumption is the core assumption of this article and also all previous research applying RL for optimal trade execution problems [3, 10, 11]. The reason is that we are training and testing on the historical data and cannot account for the permanent market impact. However, the equities we choose in the article are liquid, and the actions are relatively small compared with the market volumes. Therefore, the assumption should be reasonable.

Secondly, we ignore the commissions and exchange fees as our research is primarily aimed at institutional investors, and those fees are relatively small fractions and are negligible. Thirdly, we apply a quadratic penalty if the trading volume exceeds the available quantities of the top 5 bids. Finally, the remaining unexecuted shares will be liquidated at the last time step to ensure the execution of all the shares.

3 Experimental Results

In this section, we compare the performances of the DRL models trained on every stock data with the DRL models trained on FB only (applied to other stocks using the models trained on FB). From the perspective of learning stability, both DQN and PPO LSTM based models converge fast and have significantly outperformed the baseline models on most stocks during the backtesting. In

our experiment, we apply DeepMind’s framework to assess the stability in the training phase and the performance evaluation in the backtesting [8].

3.1 Data Sources

We use the NYSE daily millisecond TAQ data from January 1st, 2018 to December 31st, 2018, downloaded from WRDS. The TAQ data is used to reconstruct the LOB. Only the top 5 price levels from both seller and buyer sides are kept and aggregated at 5 s, which is the minimum trading interval.

3.2 Experimental Methodology and Settings

In our experiments, we apply both the DQN and PPO LSTM algorithms on 14 stocks including Facebook (FB), Google (GOOG), Nvidia (NVDA), Msci (MSCI), Tesla (TSLA), PayPal (PYPL), Qualcomm (QCOM), Goldman Sachs (GS), Salesforce.com (CRM), Boeing (BA), McDonald’s Corp (MCD), PepsiCo (PEP), Twilio (TWLO), and Walmart (WMT) which cover technology, financial and retail industries. We tune the hyperparameters on FB only and apply the same neural network architecture to the remaining stocks due to limited computing resources. The experiment follows the steps below.

1. We obtain one-year millisecond Trade and Quote (TAQ) data of 14 stocks above from WRDS and reconstruct it into the LOB. Then, we split the data into training (January–September) and test sets (October–December). We set the execution horizon to be 1 min, and the minimum trading interval to be 5 s.
2. The hyperparameters⁶ are tuned on FB only due to the limited computing resources. After fine-tuning, we apply the PPO architecture and hyperparameters to the other stocks.
3. Upon the completion of training, we check the average episode rewards’ progression. We then apply the learned policies to the testing data and compare the average episode rewards against TWAP, VWAP, and AC models. We also compare the performances of DRL models trained on every stock and DRL models trained on FB stock only.

3.3 Algorithms

TWAP: The shares are equally divided across time.

VWAP: The shares are traded at a price which closely tracks the VWAP [5].

AC: AC is defined in [1]. For a fair comparison with AC, we set its permanent price impact parameter to 0.

DQN (Lin2019): The modified DQN algorithm proposed by Lin and Beling [6].

PPO LSTM (Lin2020): The PPO algorithm uses LSTM to extract temporal patterns [7].

⁶ Please refer to the appendix for the chosen hyperparameters.

3.4 Main Evaluation and Backtesting

To evaluate the performance of the trained DRL algorithms, we apply them to the test samples from October 2018 to December 2018 for all the 14 stocks and compare their performances with TWAP, VWAP, and AC model. We report the mean of $\Delta IS = IS_{Model} - IS_{TWAP}$ (in US dollars), standard deviation of IS_{Model} . Additionally, we also report gain-loss ratio (GLR) below.

$$GLR = \frac{E[\Delta IS | \Delta IS > 0]}{E[-\Delta IS | \Delta IS < 0]} \quad (6)$$

The statistical results for all the stocks are summarized in Table 2. We observe that the DQN and PPO LSTM algorithms outperform the other models most of the time, while maintaining relative smaller standard deviations. In the experiments, we also find out that both DQN and PPO LSTM models only trained on FB perform reasonably well. Although their performances are not as good as the models trained on each stock most of the time, they reduce the risk of overfitting issue and have performed significantly better on some stocks such as MSCI (DQN), GOOG, GS, and MCD (PPO LSTM) which we highlight in bold font. It's worth mentioning that both DQN and PPO LSTM's learning curves on the above stocks (MSCI, GOOG, GS, and MCD) have converged, which suggests that learning curves are not always helpful to predict model performances on new data.

4 Conclusion and Future Work

In this article, we investigate the robustness of two DRL based trade execution models. In our experiments, we have demonstrated that both DRL models outperform TWAP, VWAP, AC models. We have also shown that both DQN and PPO LSTM models are robust and can perform well on other stocks while trained on FB data only. The experimental results are significant and indicate that the DRL model can learn essential dynamics which govern the process of trade execution and generalize well on new stocks. It also presents an opportunity for us to develop sample efficient autonomous trade execution systems in the future with the potential of significantly reducing the costs spent on data acquisition and computing.

Appendix

Hyperparameters

We fine-tuned the hyperparameters on FB only, and we did not perform an exhaustive grid search on the hyperparameter space, but rather to draw random samples from the hyperparameter space due to limited computing resources. Finally, we choose the hyperparameters listed in Tables 3 and 4.

Table 2. Model performances comparison. Mean is based on Δ IS (in US\$) and STD is based on IS (in US\$). Mean1, STD1, and GLR1 are the performance metrics for models trained on every ticker; while Mean2, STD2, and GLR2 are for models trained on FB only and are applied to other tickers.

Ticker	Mean						
	TWAP	VWAP	AC	DQN (Lin2019)		PPO LSTM	
				Mean1	Mean2	Mean1	Mean2
BA	0.00	49.66	-0.78	16.21	6.21	72.53	55.61
CRM	0.00	-120.89	-4.08	9.05	17.17	101.87	80.44
FB	0.00	-4040.78	-40.26	54.08	54.08	128.23	128.23
GOOG	0.00	-65.41	-0.92	28.43	28.18	18.06	117.73
GS	0.00	-12.58	-0.16	2.40	1.16	-1.81	9.10
MCD	0.00	-7.89	-0.13	11.86	3.63	-178.71	43.68
MSCI	0.00	34.94	-0.34	-770.24	6.78	35.50	25.57
NVDA	0.00	-31.46	-1.32	19.50	6.17	71.32	34.96
PEP	0.00	-77.54	-0.34	38.45	35.38	76.78	106.84
PYPL	0.00	-469.61	-7.84	47.35	21.82	114.38	99.54
QCOM	0.00	-6737.10	-127.87	152.02	59.59	321.43	272.52
TSLA	0.00	-63.55	-0.88	3.30	5.35	46.96	41.26
TWLO	0.00	-41.27	-2.96	4.63	4.67	21.94	21.55
WMT	0.00	-1376.78	-2.24	99.56	49.29	153.58	217.73
Ticker	Standard deviation						
	TWAP	VWAP	AC	DQN (Lin2019)		PPO LSTM	
				STD1	STD2	STD1	STD2
BA	335.05	503.64	335.80	326.62	331.10	294.09	285.60
CRM	836.69	1344.44	849.32	830.79	835.70	859.73	822.37
FB	4154.47	14936.29	4517.32	4,117.55	4,117.55	4,462.66	4,462.66
GOOG	707.03	1394.18	707.40	696.35	692.24	690.63	636.15
GS	92.74	190.89	93.19	91.07	92.02	113.71	85.94
MCD	221.72	420.70	222.10	214.50	219.36	576.29	181.03
MSCI	159.14	220.25	159.59	642.64	156.64	145.73	144.77
NVDA	478.18	726.23	479.65	454.73	468.30	431.11	452.77
PEP	1058.16	2389.84	1058.48	1,033.27	1,048.05	997.11	991.28
PYPL	672.69	2580.74	756.79	631.29	654.51	624.24	584.96
QCOM	2913.72	27131.03	4724.56	2,821.33	2,875.79	2,654.01	2,664.37
TSLA	224.02	488.35	227.79	226.80	223.35	183.42	190.96
TWLO	196.88	341.32	215.42	196.34	194.60	186.11	185.21
WMT	1442.92	6699.03	1456.07	1,386.26	1,416.75	1,355.22	1,321.25
Ticker	GLR						
	TWAP	VWAP	AC	DQN (Lin2019)		PPO LSTM (Lin2020)	
				GLR1	GLR2	GLR1	GLR2
BA	-	0.59	0.04	1.47	1.44	1.54	1.94
CRM	-	0.50	0.02	1.18	1.25	1.43	1.41
FB	-	0.16	0.00	1.04	1.04	0.63	0.63
GOOG	-	0.46	0.10	1.28	1.87	1.34	2.11
GS	-	0.46	0.16	1.40	1.16	0.58	1.05
MCD	-	0.49	0.11	2.00	1.26	0.23	3.24
MSCI	-	0.72	0.15	0.09	2.08	1.55	2.35
NVDA	-	0.51	0.06	1.54	1.25	1.21	0.91
PEP	-	0.40	0.18	1.49	1.49	1.30	1.64
PYPL	-	0.22	0.01	2.13	1.43	1.64	2.52
QCOM	-	0.08	0.01	1.53	1.06	1.63	1.86
TSLA	-	0.41	0.11	1.27	1.26	1.35	2.38
TWLO	-	0.41	0.05	1.11	1.37	1.49	1.67
WMT	-	0.17	0.02	1.94	1.35	1.32	2.23

Table 3. Hyperparameters for DQN (Lin 2019).

Hyperparameter	DQN (Lin2019)	Description
Minibatch size	16	Size of a batched sampled from replay buffer for training
Replay memory size	10000	Size of the replay buffer
Target network update frequency	2000	Update the target network every ‘target network update freq’ steps
Sample batch size	4	Update the replay buffer with this many samples at once
Discount factor	0.99	Discount factor gamma used in the Q-learning update
Timesteps per iteration	100	Number of env steps to optimize for before returning
Learning rate	0.0001	The learning rate used by Adam optimizer
Initial exploration	0.15	Fraction of entire training period over which the exploration rate is annealed
Final exploration	0.05	Final value of random action probability
Final exploration frame	125000	Final step of exploration
Replay start size	5000	The step that learning starts and a uniform random policy is run before this step
Number of hidden layers	2 for both main and noisy networks	All of main, dueling and noisy networks have 2 hidden layers
Number of hidden nodes	[128,128] for main network and [32,32] for noisy network	The main and dueling networks have 128 hidden nodes for each hidden layer; while noisy network has 32 hidden nodes for each hidden layer
Number of input/output nodes	Input: 51; output: 21	The input layer has 51 nodes while the output layer has 21 nodes
Activation functions	Relu for hidden layers and linear for output layer	For the hidden layers, we use Relu activation functions and a linear function for the output layer

Table 4. Hyperparameters for PPO LSTM (Lin2020).

Hyperparameter	PPO LSTM (Lin2020)
Minibatch size	32
Sample batch size	5
Train batch size	240
Discount factor	1
Learning rate	Linearly annealing between 5e-5 and 1e-5
KL coeff	0.2
VF loss coeff	1
Entropy coeff	0.01
Clip param	0.2
Hidden layers	2 hidden layers with 128 hidden nodes each
Activation functions	Relu for hidden layers and linear for output layer
# input/output nodes	Input: 22; output: 51
Maximum sequence length	12
LSTM cell size	128

Training and Stability

Assessing the stability and the model performance in the training phase is straightforward in supervised learning by evaluating the training and testing samples. However, it is challenging to evaluate and track the RL agent's progress during training, since we usually use the average episode rewards gained by the agent over multiple episodes as the evaluation metric to track the agent's learning progress. The average episode reward is usually noisy since the updates on the parameters of the policy can seriously change the distribution of states that the DRL agent visits.

In Fig. 2, we observe that the DRL agents converge fast in less than 200,000 steps⁷. The PPO+LSTM converge to higher average IS values on most stocks, while having shorter trading lengths. It indicates that not only do they reduce the execution costs, but also they trade more quickly to avoid timing risks.

⁷ Odd index represents the learning curves for the IS, while even index represents for the learning curves for the total # of steps to complete the trades.

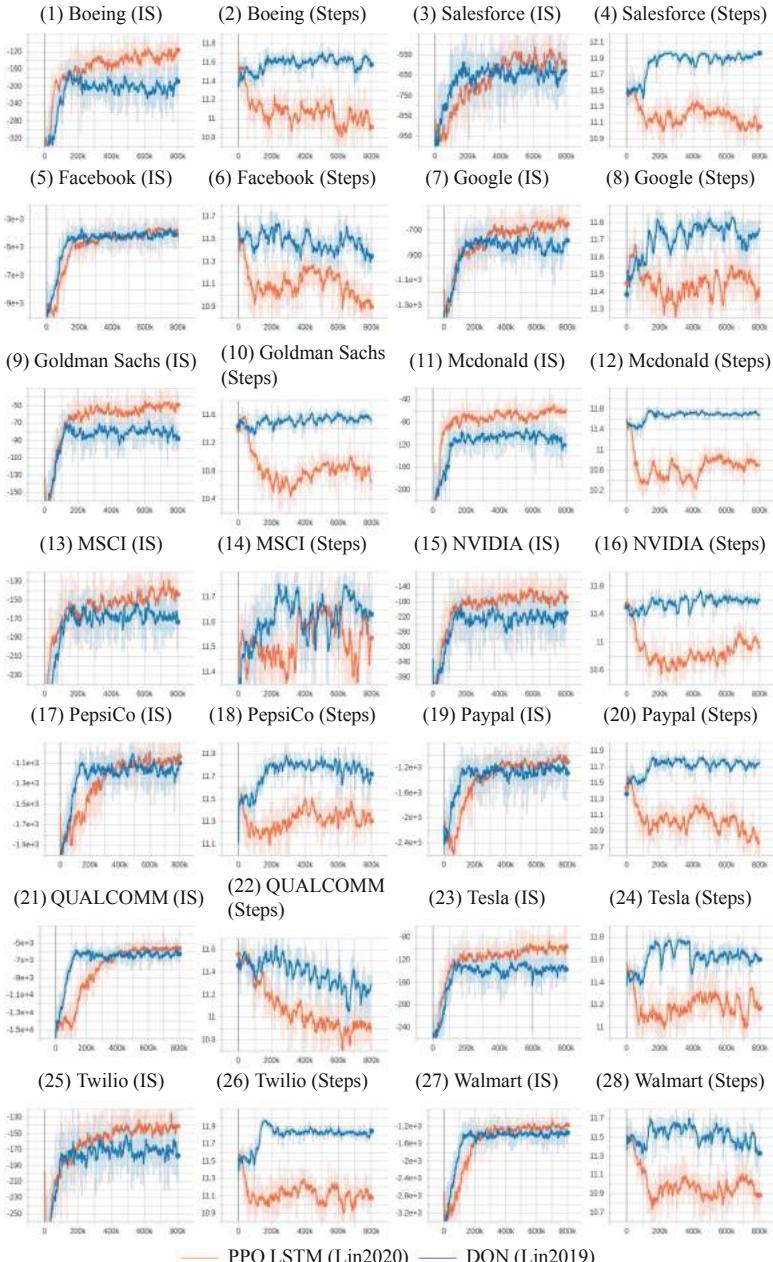


Fig. 2. Training Curves Tracking the DRL Agent's Average Implementation Shortfalls in US\$ (y-Axis for odd #) and Average Trading Steps Per Episode (y-axis for Even #) Against Steps (x-Axis): a. Each Point is the Average IS Per Episode; b. Average Trading Steps Per Episode.

References

1. Almgren, R., Chriss, N.: Optimal execution of portfolio transactions. *J. Risk* **3**, 5–40 (2000)
2. Fortunato, M., Azar, M.G., Piot, B., Menick, J., Osband, I., Graves, A., Mnih, V., Munos, R., Hassabis, D., Pietquin, O., Blundell, C.: Noisy networks for exploration. In: ICLR (2018)
3. Hendricks, D., Wilcox, D.: A reinforcement learning extension to the Almgren-Chriss framework for optimal trade execution. In: Proceedings from IEEE Conference on Computational Intelligence for Financial Economics and Engineering, pp. 57–464, London, UK. IEEE, March 2014
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
5. Kakade, S.M., Kearns, M., Mansour, Y., Ortiz, L.E.: Competitive algorithms for vwap and limit order trading. In: Proceedings of the ACM Conference on Electronic Commerce, New York, NY (2004)
6. Siyu, L., Beling, P.A.: Optimal liquidation with deep reinforcement learning. In: 33rd Conference on Neural Information Processing Systems (NeurIPS): Deep Reinforcement Learning Workshop, p. 2019. Vancouver, Canada (2019)
7. Lin, S., Beling, P.A.: An end-to-end optimal trade execution framework based on proximal policy optimization. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, pp. 4548–4554. International Joint Conferences on Artificial Intelligence Organization, July 2020
8. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602* (2013)
9. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015)
10. Nevmiyaka, Y., Feng, Y., Kearns, M.: Reinforcement learning for optimal trade execution. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 673–68, Pittsburgh, PA, June 2006. Association for Computing Machinery (2006)
11. Ning, B., Lin, F.H.T., Jaimungal, S.: Double deep q-learning for optimal execution. *arXiv:1812.06600* (2018)
12. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. *arXiv:1811.08540v2* (2017)
13. Sutton, R., Barto, A.: Reinforcement Learning: An Introduction. MIT Press, Cambridge (1998)
14. van Hasselt, H.: Double q-learning. In: Advances in Neural Information Processing Systems, pp. 2613–2621. Vancouver, British Columbia, Canada, December 2010
15. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: AAAI Conference on Artificial Intelligence (2016)
16. Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., Freitas, N.: Dueling network architectures for deep reinforcement learning. In: Proceedings of the 33rd International Conference on Machine Learning, pp. 1995–2003, New York, NY. JMLR, June 2016



On Fairness in Voting Consensus Protocols

Sebastian Müller¹, Andreas Penzkofer², Darcy Camargo², and Olivia Saa^{2(✉)}

¹ Aix Marseille Université, CNRS, Centrale Marseille, I2M - UMR 7373,
13453 Marseille, France

sebastian.muller@univ-amu.fr

² IOTA Foundation, 10405 Berlin, Germany
{andreas.penzkofer,darcy.camargo,olivia.saa}@iota.org

Abstract. Voting algorithms have been widely used as consensus protocols in the realization of fault-tolerant systems. These algorithms are best suited for distributed systems of nodes with low computational power or heterogeneous networks, where different nodes may have different levels of reputation or weight. Our main contribution is the construction of a *fair* voting protocol in the sense that the influence of the eventual outcome of a given participant is linear in its weight. Specifically, the fairness property guarantees that any node can actively participate in the consensus finding even with low resources or weight. We investigate effects that may arise from weighted voting, such as centralization, loss of anonymity, scalability, and discuss their relevance to protocol design and implementation.

Keywords: Fairness · Voting consensus protocols · Heterogeneous network · Sybil attack

1 Introduction

1.1 Preliminaries

Weighted voting in distributed systems increases efficiency and network reliability, but it also raises additional risks. It deviates from the *one node - one vote*-principle to allow a defense against Sybil attacks if the weight corresponds to a *scarce resources* or recurring costs and fee, e.g. [1]. However, weighted voting may induce a loss of anonymity for the nodes and may incentivize centralization.

In most of the systems that are based on weighted voting, all participants are allowed to vote, and a centralized entity counts the votes and takes a decision. In the decentralized setting, participants or nodes have to find consensus on the outcome of the vote. In the case where the number of nodes is high, and not all members can communicate with all other nodes, protocols where nodes only sample a certain number of nodes can be used to aggregate information, [2], or find consensus, [3–9].

This article focuses on a certain class of voting consensus protocols, namely, binary majority consensus. Some basic algorithms in this protocol class are simple majority consensus, [2], Gacs-Kurdyumov-Levin [3], and random neighbors majority, [7]. The basic idea of these majority voting protocols is that nodes query other nodes about their current opinion, and adjust their own opinion throughout several rounds based on the proportion of other opinions they have observed.

Voting protocols have been successfully applied in a wide range of engineering and economic applications [10–12], and lead to the emerging science of socio-physics [13]. Recently, [8] introduced the fast probabilistic consensus (FPC) protocol, an amelioration of the classical consensus voting protocols that is efficient and robust in Byzantine infrastructure. This protocol is studied in more detail in [9] and serves this note as a reference for a voting consensus protocol.

1.2 Weights and Fairness

The main contribution of this work is to define an adaption of the majority consensus protocol to a setting that allows nodes to have different weights. We define *voting power*, Sect. 4, as a power index to describe the influence of each node on the outcome of the consensus protocol. In comparison with standard voting models where the whole population is sampled, we have two instances where the weights may come in: firstly, in the sampling of the nodes to query and secondly, in the weighting of the votes to apply the majority rule. An essential consequence of fairness of the protocol is that it allows defense against Sybil attacks. Protection against Sybil attacks is especially needed in permissionless settings, where a malicious actor may gain a disproportionately large influence on the voting by creating a large number of identities.

Moreover, fairness allows even nodes with very few resources or weight to participate in the consensus finding. This property is particularly important in networks where the sum of the weights of nodes with small weight is considerable.

Besides technical and economic considerations, we want to mention the possible social impacts of an unfair system. For instance, unfair situations can make participants of the network unhappy, and this should be a real consideration in judging the efficiency and adaption of a protocol, e.g. [14]. In particular, this is of importance in community-driven projects such as IOTA.

1.3 IOTA

IOTA is an open-source distributed ledger and cryptocurrency designed for the Internet of things. For the next generation of the IOTA protocol [15] introduces *mana* in various places in order to obtain fairness. For instance, the protocol uses mana to obtain fairness in rate control, [16, 17], where an adaptive PoW-difficulty guarantees that any node, even with low hashing power, can achieve similar throughput for given mana. This note discusses how mana is used as a weight in FPC to construct a fair consensus protocol.

1.4 Outline

The rest of the paper organizes as follows. After giving an overview of previous work in Sect. 2, we give a brief introduction to voting consensus protocol and FPC in Sect. 3. The main contribution is the formulation of a proper mathematical framework in Sect. 4 and the construction of a weighted voting consensus protocol that is *fair* in the sense that the voting power is proportional to the weights of the nodes. Section 5 proposes to use a Zipf law for modeling the weight distribution. Under the validity of the Zipf law, we discuss in Sect. 6 impacts of the weighted voting on scalability and implementation of the protocol. Section 7 present some simulation results that show the behavior of the protocol in Byzantine infrastructure for different degrees of centralization of the weights. We conclude in Sect. 8 with a discussion.

2 Related Work

2.1 Weighted Voting Consensus Protocols

Voting consensus protocols (without weights) are widely studied in theory and applications, and they play an important role in social learning. Also, weighted voting systems have a long history in election procedures, and often one is interested in measuring the influence of the power of the given participants, [18]. Despite these facts, we were not able to find related results on voting consensus protocols with weights. Recently, [19] describes a model that is related to FPC, and that considers biased or stubborn agents. For the sake of brevity, we refer to [8,9] for more details on related work and references therein.

2.2 Fairness

Fairness plays a prominent role in many areas of science and applications. It is, therefore, not astonishing that it plays its part also in DLT. For instance, PoW in Nakamoto consensus ensures that the probability of creating a new block is proportional to the computational power of a node; see [20] for an axiomatic approach to block rewards and further references. In PoS-based blockchains, the probability of creating a new block is usually precisely proportional to the node's balance. However, this does not always have to be the *optimal* choice, [21,22].

3 Voting Consensus Protocols

We give a brief definition of binary voting protocols in this section. More details can be found, for instance, in [3–7]. We refer to [8,9] for more information on the fast probabilistic consensus (FPC).

To define the protocol accurately, we introduce some notation. We consider network with N nodes that are indexed by $1, 2, \dots, N$. We assume that every node can query any other node. This assumption is made for the sake of a better

presentation and not necessary. In fact, simulation studies [7, 9] show that it is sufficient if every node is able to query about half of the other nodes. It also seems to be a reasonable assumption because nodes with high weights will likely be known to every participant in the network. Every node i has an opinion or state. The opinion of a node i at time t is denoted by $s_i(t)$, and takes values in $\{0, 1\}$. Every node i starts with an initial opinion $s_i(0)$.

At each (discrete) time step a node i chooses k random nodes $C_i = C_i(t)$ and queries their opinions. Denote by $k_i(t) \leq k$ the number of replies received by node i at time t . If the reply from j is not received in due time we set $s_j(t) = 0$. The updated mean opinion is then

$$\eta_i(t+1) = \frac{1}{k_i(t)} \sum_{j \in C_i} s_j(t).$$

Note that repetitions are possible since the sample C_i of a node i is chosen using sampling with replacement.

We consider a basic version of FPC introduced in [9]. Specifically, we remove the cooling phase of FPC and the randomness of the initial threshold τ . We consider i.i.d. random variables U_t , $t = 1, 2, \dots$, with law $\text{Unif}([\beta, 1 - \beta])$, for some $\beta \in [0, 1/2]$. The update rule for the opinion of a node i is then given by

$$s_i(1) = \begin{cases} 1, & \text{if } \eta_i(1) \geq \tau, \\ 0, & \text{otherwise,} \end{cases}.$$

For subsequent rounds, i.e. $t \geq 1$:

$$s_i(t+1) = \begin{cases} 1, & \text{if } \eta_i(t+1) > U_t, \\ 0, & \text{if } \eta_i(t+1) < U_t, \\ s_i(t), & \text{otherwise.} \end{cases}$$

Note that if $\tau = \beta = 0.5$, the protocol reduces to a standard majority consensus. An asymmetric choice of τ , $\tau \neq 0.5$ allows the protocol the distinction between the two kinds of integrity failure, [8, 9]. It is important, that the sequence of random variables U_t are the same for all nodes. A more detailed discussion on the use of decentralized random number generators is given in [9].

In contrast to many theoretical papers on majority dynamics, a local termination rule is needed for practical applications. Every node keeps a variable `cnt` that is incremented by 1 if there is no change in its opinion. The variable is set to 0 if there is a change of opinion. The node considers the current state as final if the counter reaches a certain threshold 1, i.e., $\text{cnt} \geq 1$. In the absence of autonomous termination the protocol stops after `maxIt` iterations.

4 Fairness

In this section, we propose a proper mathematical framework of fairness. The weight of the N nodes is given by $\{m_1, \dots, m_N\}$ with $\sum_{i=1}^N m_i = 1$. In the sampling of the queries a node j is chosen with probability

$$p_j = \frac{f(m_j)}{\sum_{i=1}^N f(m_i)}. \quad (1)$$

Each opinion $s_j(t)$ is weighted by $g_j = g(m_j)$:

$$\eta_i(t+1) = \frac{1}{\sum_{j \in \mathcal{C}_i} g_j} \sum_{j \in \mathcal{C}_i} g_j s_j(t). \quad (2)$$

The other parts of the protocol remain unchanged. Since we sample with replacement we count the number of times a node i is chosen and denote this number by y_i . Using the fact that the sampling can be expressed by a multinomial distribution we can calculate the expected result of a query as

$$\mathbb{E}\eta(t+1) = \sum_{i=1}^N s_i(t) v_i, \quad (3)$$

where

$$v_i = \sum_{\mathbf{y} \in \mathbb{N}^N : \sum y_i = k} \frac{k!}{y_1! \cdots y_N!} \frac{y_i g_i}{\sum_{n=1}^N y_n g_n} \prod_{j=1}^N p_j^{y_j}. \quad (4)$$

We call v_i the *voting power* of node i . This quantity measures the influence of the node i . We would like the voting power to be proportional to the weight since this would induce a robustness to splitting and merging, [23].

Definition 1 (Robust to Splitting and Sybil Attacks). A voting scheme is robust to Sybil attacks if a node i splits into nodes i_1 and i_2 with a weight splitting ratio $x \in (0, 1)$, then

$$v_i(m_i) \geq v_{i_1}(xm_i) + v_{i_2}((1-x)m_i). \quad (5)$$

Definition 2 (Robust to Merging). A voting scheme is robust to merging if a node i splits into nodes i_1 and i_2 with a weight splitting ratio $x \in (0, 1)$, then

$$v_i(m_i) \leq v_{i_1}(xm_i) + v_{i_2}((1-x)m_i). \quad (6)$$

Definition 3 (Fairness). A voting scheme (f, g) is fair if it is robust to Sybil attacks and robust to merging. In other words, if a node i splits into nodes i_1 and i_2 with a weight splitting ratio $x \in (0, 1)$, then

$$v_i(m_i) = v_{i_1}(xm_i) + v_{i_2}((1-x)m_i). \quad (7)$$

In the case where $g \equiv 1$ we construct a voting scheme that is fair for all possible choices of k and weight distributions.

Theorem 1. Let us assume that $g \equiv 1$. Then, the voting scheme (f, g) is fair if and only if f is the identity function $f = id$.

Proof. We consider $g \equiv 1$. In this case we can simplify (7) to

$$v_i = \frac{1}{k} \sum_{\mathbf{y} \in \mathbb{N}^N : \sum y_i = k} y_i \mathbb{P}[\mathbf{y}],$$

where \mathbf{y} follows a multinomial distribution using the probability vector $\{p_j\}$ and making k selections. Hence, we obtain

$$v_i(m_i) = p_i = \frac{f(m_i)}{\sum_{j=1}^N f(m_j)}. \quad (8)$$

Now, let $S = \sum_{j=1}^N f(m_j)$ and

$$\Delta_x = f(xm_i) + f((1-x)m_i) - f(m_i). \quad (9)$$

The fairness condition (7) becomes

$$\frac{f(m_i)}{S} = \frac{f(xm_i)}{S + \Delta_x} + \frac{f((1-x)m_i)}{S + \Delta_x}, \quad \forall x \in (0, 1), \quad (10)$$

which is equivalent to

$$\Delta_x f(m_i) = S \Delta_x, \quad \forall x \in (0, 1). \quad (11)$$

This means that either $f(m_i) = S$, meaning that there is only one node, or that $\Delta_x \equiv 0$, meaning that for all $x \in (0, 1)$ we have that

$$f(m) = f(xm) + f((1-x)m), \quad (12)$$

and that f is a linear function.

On the other hand, if the nodes are queried at random without weighting the probability for selection, i.e., $f \equiv 1$, then there exists no voting scheme that is fair for all k .

Theorem 2. *For $f \equiv 1$ there exists no voting scheme (f, g) with for all $g_i > 0$ that is fair for all k .*

Proof. In the case of $f \equiv 1$ the voting power simplifies to

$$v(m_i) = \frac{1}{N^k} \sum_{\mathbf{y} \in \mathbb{N}^N : \sum y_i = k} \frac{k!}{y_1! \cdots y_N!} \frac{y_i g_i}{\sum_{n=1}^N y_n g_n}. \quad (13)$$

We will consider a special situation that does not satisfy the fairness condition. We consider the situation with $N = 2$ nodes and $m_1 = 2/3$ and $m_2 = 1/3$. The voting power of the first node is then

$$v\left(\frac{2}{3}\right) = \mathbb{E} \left[\frac{X g(\frac{2}{3})}{X g(\frac{2}{3}) + (k-X) g(\frac{1}{3})} \right], \quad (14)$$

where X follows a binomial distribution $\mathcal{B}(k, 1/2)$. Now, it suffices to show that the right hand side of Eq. (14) is not constant in k . Due to the law of large numbers and the dominated convergence theorem we have that

$$\mathbb{E} \left[\frac{X g(\frac{2}{3})}{X g(\frac{2}{3}) + (k-X) g(\frac{1}{3})} \right] \xrightarrow{k \rightarrow \infty} \frac{g(\frac{2}{3})}{g(\frac{2}{3}) + g(\frac{1}{3})}. \quad (15)$$

We conclude by observing that

$$\mathbb{E} \left[\frac{Xg(\frac{2}{3})}{Xg(\frac{2}{3}) + (k - X)g(\frac{1}{3})} \right] < \frac{g(\frac{2}{3})}{g(\frac{2}{3}) + g(\frac{1}{3})}, \quad (16)$$

where the strict inequality follows from Jensen's Inequality using that X is not a constant almost surely and $g(\frac{2}{3}) > g(\frac{1}{3})$.

For these reasons we fix from now on $g \equiv 1$ and $f = id$.

5 Distribution of Weight

The existence of a fair voting scheme is independent of the actual distribution of weight. However, to make a more detailed prediction of the qualities of the voting consensus protocol, it may be appropriate to make assumptions on the weights.

Probably the most appropriate modelings of the weight distributions rely on universality phenomena. The most famous example of this universality phenomenon is the central limit theorem. While the central limit is suited to describe statistics where values are of the same order of magnitude, it is not appropriate to model more heterogeneous situations where the values might differ in several orders of magnitude. For instance, values in most (crypto-)currency systems are not distributed equally; [24].

Zipf's law describes mathematically various models of heterogeneous real-world systems. For instance, many economic models, [25], use these laws to describe the wealth and influence of participants. This makes the Zipf law a natural candidate for modeling the weight distribution. It can be defined as follows. The n th largest value $y(n)$ follows approximately a power law, i.e. it should behave roughly as

$$y(n) = Cn^{-s} \quad (17)$$

for the first few $n = 1, 2, 3, \dots$ and some parameters $s > 0$ and normalising constant C . We refer to [26] for an excellent introduction to this topic.

A first and convenient way to verify a Zipf law is to plot the data on a log-log graph, i.e. the axes being $\log(\text{rank order})$ and $\log(\text{value})$. A linear plot is then a strong indication for a Zipf law. An estimation of the parameter s can be given by a linear regression. As an example, we show the distribution of the top 10.000 richest IOTA addresses together with a fitted Zipf law in Fig. 1.

Due to the universality phenomenon mentioned above and the observation made in Fig. 1, we assume, in the following sections, that the weight distribution follows a Zipf law.

6 Scalability and Message Complexity

An essential property of voting consensus protocols is their scalability. In fact, at every round, every node is queried on average k times indifferent to the

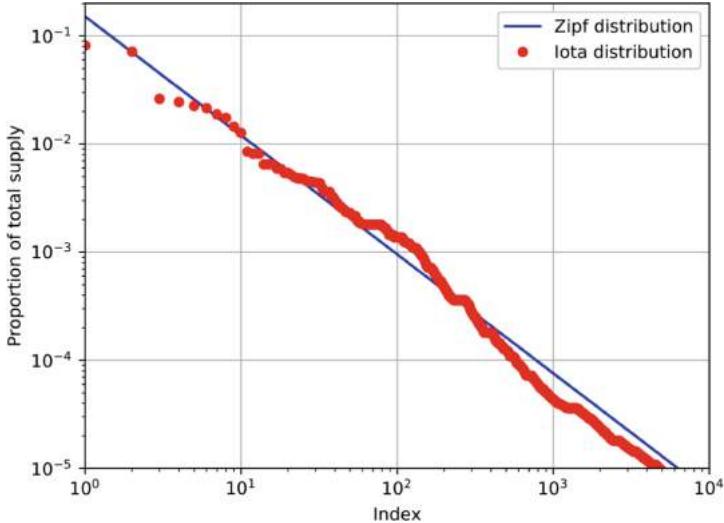


Fig. 1. The top 10.000 richest IOTA addresses together with a fitted Zipf law with $s = 1.1$; as of July 2020.

network size. In our proposed fair voting schemes where nodes are sampled proportional to their weight, this is no longer true, and nodes with higher weight are queried more often. This affects the scalability of the protocol and might generate feedback on the weight distribution.

Lemma 1. *We assume the weights to follow a Zipf law with parameters (s, N) . Then, the average number of queries a node of rank $h(N)$ receives per round is of order (as $N \rightarrow \infty$)*

1. $\Theta(N^s h(N)^{-s})$, if $s < 1$;
2. $\Theta(\frac{N}{\log N} h(N)^{-1})$, if $s = 1$;
3. $\Theta(N h(N)^{-s})$, if $s > 1$.

Proof. At every round a node of rank $h(N)$ is queried on average

$$N \cdot \frac{h(N)^{-s}}{\sum_{n=1}^N n^{-s}} \quad (18)$$

times. In the case $s < 1$ this expression becomes asymptotically $\Theta(N^s h(N)^{-s})$. In the case $s = 1$ we obtain $\Theta(\frac{N}{\log N} h(N)^{-1})$, and if $s > 1$ it becomes $\Theta(N h(N)^{-s})$.

Let us note that, if $s > 0$ the node with the highest weight is queried $\Theta(N^s)$, $\Theta(\frac{N}{\log N})$, or $\Theta(N)$ times. Nodes with considerable lower weight, i.e. of rank $\Theta(N)$, are queried on average only $\Theta(1)$ times. This is in contrast to the

case $s = 0$ where every node has the same weight, and every node is queried on average a constant number of times.

Therefore, nodes with high weights have an incentive to communicate their opinions through different communication channels, e.g. to gossip their opinions on an underlying peering network and not to answer each query separately. On the other hand, the situation where all nodes gossip their opinions is not optimal neither. In this case, every node would have to send $\Theta(N)$ messages. In the following we propose a criterion for the gossiping of opinions.

We assume that nodes with high weights have higher throughput than nodes with lower weights and can treat more messages than nodes with lower weights. We consider the situation where only the $\Theta(\log(N))$ highest weights nodes gossip their opinions. This leads to $\Theta(\log N)$ messages for each node in the gossip layer. The highest weight node, that does not gossip its opinions, receives on average $\Theta((\frac{N}{\log N})^s)$ messages if $s < 1$, $\Theta(\frac{N}{(\log N)^2})$ messages if $s = 1$, and $\Theta(\frac{N}{(\log N)^s})$ messages if $s > 1$. In this scenario, middle rank nodes, i.e. those of rank between $\Theta(\log N)$ and $\Theta(N)$, have the highest message load.

A possible consequence might be that these *middle rank* nodes might gossip their opinion leading to possible congestion of the network or might stop to answer queries leading to less security of the consensus protocol. A less selfish response of such a node could be to either split up the node into several nodes or to pool with other nodes to gossip their opinions according to the above threshold rule. This, however, might influence the distribution of weight.

The above situation may apply well to heterogeneous networks, i.e. the computational power and bandwidths of the node might differ in several orders of magnitudes, [16, 17]. In homogeneous networks the above issues may be solved by a *fair* attribution of message complexity.

Lemma 2. *We assume the weights to follow a Zipf law with parameters s and N . Then there exists a fair threshold for gossiping opinions such that every node has to process the same order of messages. The message load for each node is $O(\sqrt{N})$ for all choices of s .*

Proof. We have to find a threshold such that the maximal number of queries a node receives equals the number of gossiped messages. For $s < 1$ this leads to the following equation:

$$N^s h(N)^{-s} = h(N). \quad (19)$$

Hence, a threshold of order $N^{\frac{s}{s+1}}$ leads to $\Theta(N^{\frac{s}{s+1}})$ messages for every node to send. For $s > 1$ we obtain similarly a threshold of $N^{\frac{1}{1+s}}$ leading to $\Theta(N^{\frac{1}{1+s}})$ messages. For $s = 1$ we obtain a message complexity for each node of $O(\sqrt{N})$.

7 Simulations

In this section, we present a short simulation study that shows how the performance of the FPC depends on the distribution of the weights.

7.1 Threat Model

We assume that an adversary holds a proportion q of the total weight. The adversary splits its weight equally between its qN nodes. In this way, each adversary node holds $1/N$ of the total weight. We assume that the adversary is at every moment aware of all opinions and transmits at time $t + 1$ the opinion of the weighted minority of the honest nodes of step t .

7.2 Failures

Standard failures of consensus protocols are integration failure, agreement failure, and termination failure. There are different possibilities to generalize these failures to the case of heterogeneous weight distributions. Since in many applications the agreement failure is the most severe, we consider only agreement failure in this note. In the non-weighted setting an agreement failure occurs if not all nodes decide on the same opinion. However, in the weighted setting it makes sense to take the weights into account. For this reason, we consider the 1%-agreement failure in the sense that a failure occurs if nodes of at least 1% weight differ in their final decision.

7.3 Results

We choose the initial average opinion p_0 equal to the value of the first threshold τ . This can be considered as the critical case since for values of $p_0 \gg \tau$ or $p_0 \ll \tau$, the agreement failure rate is so small that numerical simulation is no longer feasible. We assign the initial opinions as follows. The highest weight nodes holding together more than p_0 of the weight are assigned opinion 1 and the remaining opinion 0. Other default parameters are: $N = 1000$ nodes, $p_0 = \tau = 0.66$, $\beta = 0.3$, $l = 10$, $\text{maxIt} = 50$.

In Fig. 2 we investigate the protocol with a small sample size, $k = 20$, and study the agreement failure rate as a function of q . We see that the performance depends on the parameter s ; the higher the centralization is the lower the agreement failures are for high q , but at the price that the protocols performs worse if q is smaller. In Fig. 3 we observe an exponential decay of the agreement failure rate in the sample size k .

The code of the simulations is open source and online available.¹

¹ <https://github.com/IOTAledger/fpc-sim>.

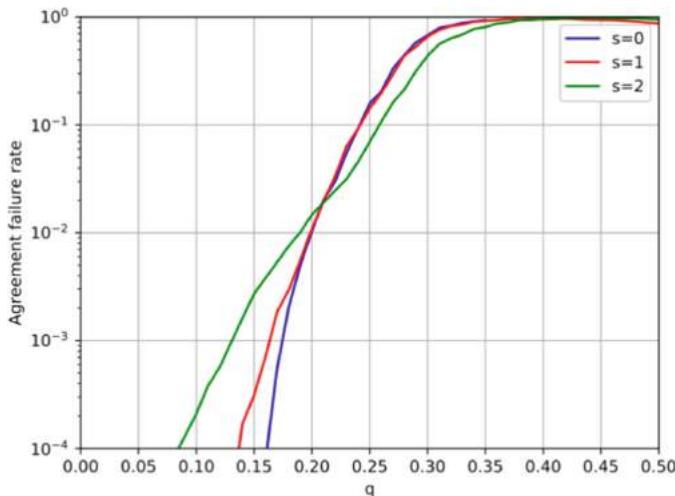


Fig. 2. Agreement Failure Rates as a Function of q for Three Different Weight Distributions and $k = 20$.

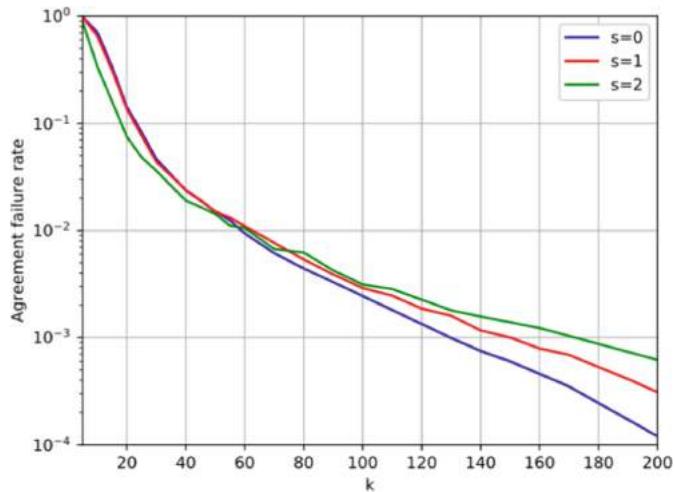


Fig. 3. Agreement failure rates with k ; $q = 0.25$.

8 Discussion

We proposed a mathematical framework to study fairness in consensus voting protocols and constructed a fair voting scheme, Sect. 4. Even though this voting scheme is robust to splitting and merging, there are *second order* effects that may incentivize nodes to optimize their weights. One example we studied concerns the message complexity and its consequences, Sect. 6. Other secondary effects that may influence the weights are, for instance; basic resource costs such as

maintaining nodes, network redundancy, and service availability. Moreover, the fact that the security of the protocol has an impact on its performance may lead to changes in the weight distribution.

Nodes with high weight are likely no longer be anonymous. In the case where these nodes gossip their opinions, and other users widely accept their reputation, some nodes may decide not longer to take part in the consensus finding but to only follow the nodes with the highest reputations. This would give disproportional weight to nodes with high weight, leading to an unfair situation.

A complete mathematical treatment of the above is not realistic; all the more the evolution of such a permissionless and decentralized consensus protocol depends also on other components of the network, as well as economic and psychological elements.

References

1. Neil, B., Shields, L.C., Margolin, N.B.: A survey of solutions to the sybil attack (2005)
2. Mossel, E., Neeman, J., Tamuz, O.: Majority dynamics and aggregation of information in social networks. *Auton. Agent. Multi-Agent Syst.* **28**, 408–429 (2014)
3. Gács, P., Kurdyumov, G.L., Levin, L.A.: One-dimensional Uniform Arrays that Wash out Finite Islands. In: *Problemy Peredachi Informatsii* (1978)
4. Moreira, A.A., Mathur, A., Diermeier, D., Amaral, L.: Efficient system-wide coordination in noisy environments. *Proc. Natl. Acad. Sci. U.S.A.* **101**, 12085–12090 (2004)
5. Kar, S., Moura, J.M.F.: Distributed average consensus in sensor networks with random link failures. In: *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP 2007*, vol. 2, pp. II–1013–II–1016, April 2007
6. Cruise, J., Ganesh, A.: Probabilistic consensus via polling and majority rules. *Queueing Syst.* **78**(2), 99–120 (2014)
7. Gogolev, A., Marchenko, N., Marcenaro, L., Bettstetter, C.: Distributed binary consensus in networks with disturbances. *ACM Trans. Auton. Adapt. Syst.* **10**, 19:1–19:17 (2015)
8. Popov, S., Buchanan, W.J.: FPC-BI: fast probabilistic consensus within byzantine infrastructures. *J. Parallel Distrib. Comput.* **147**, 77–86 (2021)
9. Capossele, A., Mueller, S., Penzkofer, A.: Robustness and efficiency of leaderless probabilistic consensus protocols within byzantine infrastructures (2019)
10. Banisch, S., Araújo, T., Louçã, J.: Opinion dynamics and communication networks. *Adv. Complex Syst.* 95–111 (2010)
11. Niu, H.-L., Wang, J.: Entropy and recurrence measures of a financial dynamic system by an interacting voter system. *Entropy* 2590–2605 (2015)
12. Przybyła, P., Sznajd-Weron, K., Tabiszewski, M.: Exit probability in a one-dimensional nonlinear q -voter model. *Phys. Rev. E* (2011)
13. Castellano, C., Fortunato, S., Loreto, V.: Statistical physics of social dynamics. *Rev. Modern Phys.* 591 (2009)
14. Rabin, M.: Incorporating fairness into game theory. Department of Economics, UC Berkeley (1991)
15. Popov, S., et al.: The coordicide (2020)

16. Vigneri, L., Welz, W., Gal, A., Dimitrov, V.: Achieving fairness in the tangle through an adaptive rate control algorithm. In: 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 146–148, May 2019
17. Vigneri, L., Welz, W.: On the fairness of distributed ledger technologies for the internet of things. In: 2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC) (2020)
18. Gambarelli, G.: Power indices for political and financial decision making: a review. *Ann. Oper. Res.* **51**(4), 163–173 (1994)
19. Arpan Mukhopadhyay, R.R., Mazumdar, R.R.: Voter and Majority Dynamics with Biased and Stubborn Agents <https://arxiv.org/abs/2003.02885> (2020)
20. Chen, X., Papadimitriou, C., Roughgarden, T.: An axiomatic approach to block rewards. In: Proceedings of the 1st ACM Conference on Advances in Financial Technologies, (New York, NY, USA), pp. 124–131. Association for Computing Machinery (2019)
21. Popov, S.: A probabilistic analysis of the Nxt forging algorithm. *Ledger* **1**, 69–83 (2016)
22. Leonardos, S., Reijnsbergen, D., Piliouras, G.: Weighted voting on the blockchain: improving consensus in proof of stake protocols (2020)
23. Leshno, J., Strack, P.: Bitcoin: an impossibility theorem for proof-of-work based protocols. Cowles Foundation Discussion Paper, no. 2204R (2019)
24. Kondor, D., Pósfai, M., Csabai, I., Vattay, G.: Do the rich get richer? An empirical analysis of the bitcoin transaction network. *PloS One* **9**, e86197 (2014)
25. Jones, C.I.: Pareto and Piketty: the macroeconomics of top income and wealth inequality. *J. Econ. Perspect.* **29**, 29–46 (2015)
26. Tao, T.: Benford's law, Zipf's law, and the Pareto distribution. <https://terrytao.wordpress.com/2009/07/03/benfords-law-zipfs-law-and-the-pareto-distribution/> (2009)



Dynamic Urban Planning: An Agent-Based Model Coupling Mobility Mode and Housing Choice. Use Case Kendall Square

Mireia Yurrita^(✉), Arnaud Grignard, Luis Alonso, Yan Zhang, Cristian Ignacio Jara-Figueroa, Markus Elkatsha, and Kent Larson

Massachusetts Institute of Technology, Cambridge, USA
mireia.yurrita@gmail.com

Abstract. As cities become increasingly populated, urban planning plays a key role in ensuring the equitable and inclusive development of metropolitan areas. MIT City Science group created a data-driven tangible platform, CityScope, to help different stakeholders, such as government representatives, urban planners, developers, and citizens, collaboratively shape the urban scenario through the real-time impact analysis of different urban interventions. This paper presents an agent-based model that characterizes citizens' behavioural patterns with respect to housing and mobility choice that will constitute the first step in the development of a dynamic incentive system for an open interactive governance process. The realistic identification and representation of the criteria that affect this decision-making process will help understand and evaluate the impacts of potential housing incentives that aim to promote urban characteristics such as equality, diversity, walkability, and efficiency. The calibration and validation of the model have been performed in a well-known geographic area for the Group: Kendall Square in Cambridge, MA.

Keywords: Agent-based modelling · Housing choice · Residential mobility · Dynamic urban planning · Pro-social city development

1 Introduction

Even if humans started to cluster around cities 5,500 years ago, it was not until about 150 years ago that cities became increasingly populated [1]. Nowadays more than 50% of the world's population lives in urban areas, which makes urban planning emerge as one of the key elements for citizens' well-being [2]. As in the following decades cities may account for 60% of the total global population, 75% of the global $C0_2$ emissions, and up to 80% of all world energy use [3], traditional systems of centralised planning might be obsolete [2].

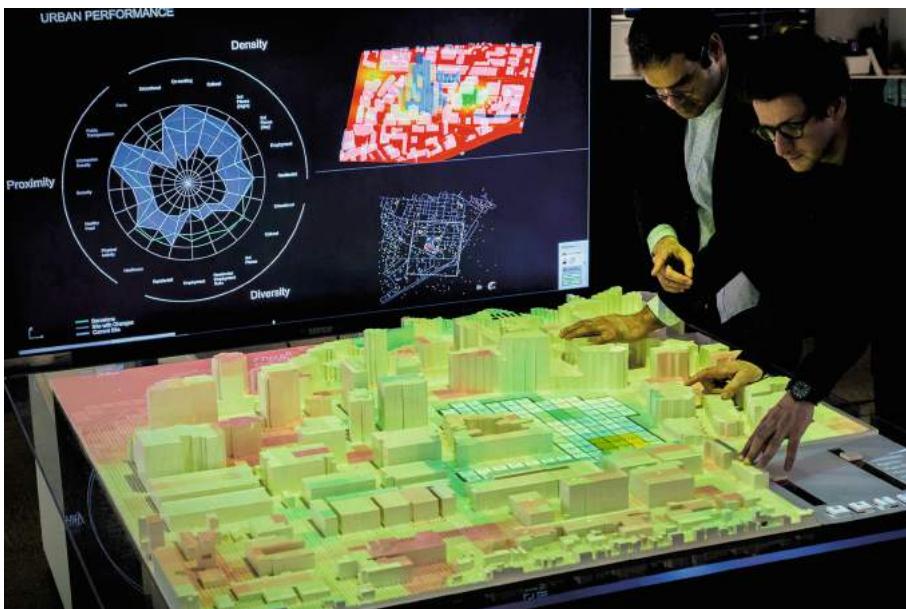


Fig. 1. CityScope Volpe: a data-driven interactive simulation tool for urban design [2]

In view of the need for more participatory decision-making processes, MIT City Science (CS) Group developed a tangible data-driven platform, CityScope. This platform brings different stakeholders together in an effort to enable a more collaborative urban design process [2,4] – Fig. 1. As part of this collective decision-making approach, the CS Group is designing a dynamic incentive system, that aims to promote pro-social urban features through dynamic incentive policies. By using data-driven agent-based simulations, the impacts on equality, diversity, walkability, and efficiency of such incentives are measured. Thanks to this information, different stakeholders, such as government representatives, urban planners, developers, and citizens, can collaboratively decide which intervention shapes the most favourable urban scenario.

Some of the incentives explored in this line of research are focused on the promotion of affordable housing within a 20-min walking distance from one's workplace. This vision is part of a broader perspective where cities are encouraged to be comprised of autonomous communities, where citizens' daily needs are met within a walking distance from their doorstep. MIT City Science intends to create the technological tools to facilitate consensus between community members, local authorities, and design professionals [2], so that cities can fulfil their ambition of having walkable districts.

This paper arises from the need to forecast citizens' criteria when choosing their homes, and the consequent mobility patterns. By doing so the effects of numerous pro-social incentives that aim to address the deficiency of affordable housing in city centres can be tested. Although some previous studies focusing on citizens' mobility choices had already been held in the group, the need to

combine mobility trends with housing-related aspects was major. In contrast to the consulted bibliography, where most approaches make use of statistical tools to respond to this joint choice research question, Agent-Based Modeling (ABM) has been deployed in this study. People's behavioral patterns have been represented based on their income profiles.

This document is organized as follows. Section 2 describes previous work on the representation of housing and residential mobility choices and highlights the benefits of using ABM when embracing the complexity of urban dynamics. Section 3 gives a detailed summary of the agents that constitute the model, the behaviours of each of them and the dynamics of the housing and mobility mode choice. Section 4 calibrates the model for the particular use case of Kendall Square using census and transportation data and Sect. 6 gives some final thoughts on the effectiveness of the method.

2 Background

2.1 Related Work

The present research is an agent-based joint representation of housing and mobility choices that intends to forecast citizens' behavioural patterns when choosing their residencies as well as to highlight impacts on mobility. This study has been built on top of previous work where different approaches to address this question have been suggested. Some of them merely focused on the housing choice search, others concentrated on the mobility mode issue and some merged both points of view.

Among the former methodologies, a two-stage behavioural housing search model [5] had been created. This model was comprised of a hazard-based "choice set formation step", and a "final residential location selection step" using a multinomial logit formulation. A theoretical housing preference and choice framework had also been given using the theory of mean-end chain [6]. The motivations that make people want to move had been studied [7] using a nested logit model, which paid special attention to how residential decisions are impacted by transportation. A multi-agent approach had also been used to model and analyze the residential segregation phenomenon [8]. And an ABM had been designed to represent the dynamics of the housing market and for the study of the effects that certain parameters, like school accessibility, would have on the outcome [9].

As far as mobility choices are concerned, some previous studies had been carried out within the MIT City Science Group using both statistical tools as well as agent-based models. The HCl platform deployed for the real-time prediction of mobility choices through discrete choice models [10] is one of the examples of how the CS Group had previously addressed the mobility choice issue. Agent-based models had also been developed in our Group to evaluate new mobility mode choices and to assess their impact on cities [11].

Among previous joint choice models, it is important to highlight a model based on the disaggregate theory where the combinations of location, housing,

automobile ownership and the commuting mode had been studied [12]. Additionally, a nested multinomial logit model to estimate the joint choice of residential mobility, and housing choice [13] had also been used.

This paper presents a novel joint choice model based on agents. Following the Schelling segregation model [14], the factors related to housing, and those related to transportation will be deployed to determine if a certain agent is willing to move or not.

2.2 Agent-Based Simulations

Most of the related work mentioned prior in this paper make use of statistical tools to study the housing and residential mobility choice phenomenon. Our research, on the contrary, will artificially reproduce the dynamics of the urban scenario using agent-based simulations. Although several platforms dedicated to agent-based modelling have been created lately, this model has been developed using the GAMA platform. GAMA allows modellers to create complex multi-level spatially explicit simulations where GIS data can be easily integrated. It also provides a high-level agent-oriented language, GAML, as well as an integrated development environment [15, 16]. The behaviour of each ‘people agent’ has been developed using this tool and some agent-specific functions have been designed in order to recreate their decision-making reasoning while representing the complexity of the surrounding dynamics.

3 Model Description

This model aims to represent the criteria adopted by citizens when choosing their residential location and mobility mode, as well as the importance given to each of these parameters. The realistic characterization of these behaviours enables the study of the consequences that certain housing incentives and urban disruptions might entail.

3.1 Entities, State Variables, and Scales

The **environmental Variables** used to shape the urban scenario and represent its dynamics include:

- **Census Block Group:** polygon representing the combination of census blocks, smallest geographic areas for which the US Bureau of Census collects data [17]. Each census block group will be formed by the following attributes: *GEOID*: unique geographical ID for each census block group, *vacant_spaces*: number of vacant housing options available within the boundaries of this polygon (previously a web scraping process has been held and the availability of accommodation and its price identified [18] within the area of interest), *city*: broader unit the block group belongs to [17], *rent_vacancy*: mean rent for each housing option based on online rent information [18], *population*: map

indicating the number of citizens of each income profile that actually live in the area [19], *has_T*: boolean attribute pointing out whether the block group has a T station or not [20] and *has_bus*: boolean attribute that indicates if bus services are available in the area [20]. Census block groups have been the environmental variables chosen to represent the broadest granularity in the model, since applying the same unit as the US Census Bureau enables the calibration and validation of the model using census data.

- **Building:** polygon representing a finer granularity for the area where the urban interventions are considered to take place. The aforementioned census block groups are the alternative to living in a building within the area of interest. The attributes of ‘building’ agents include: *associated_block_group*: block group in which the building is located [17], *vacant_spaces*: number of vacant dwelling units within the building [18] and *rent_vacancy*: mean rent for each housing option within the building [18].
- **Road:** network of roads that agents can use to move around. *mobility_allowed* is the only attribute of the road network, which is taken into account when considering different commuting mobility options and when the resulting time for each of them is calculated [17].

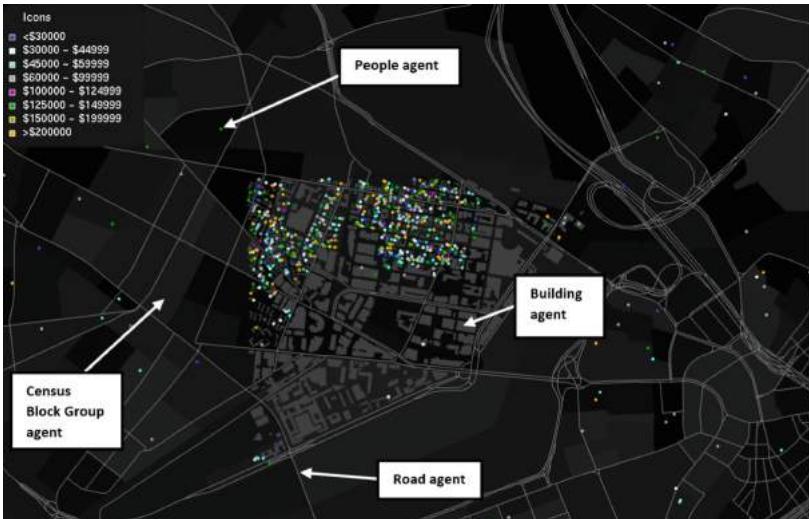


Fig. 2. Model overview, where people, building, road and census block group – areas with different transparency – agents have been illustrated for Kendall Square in Cambridge, MA.

As far as **agents** are concerned, ‘people agents’ are defined the following way:

- **People:** citizens who work in the area of interest and whose objective is to find the most appropriate housing and mobility option based on their criteria.

Their attributes include: *type*: type of ‘people agent’ based on their annual income profile [19], *living_place*: building or census block group where they decide to live in each iteration of the searching process, *activity_place*: building in the area of interest where each agent works, *possible_mobility_modes*: list of mobility modes that are available for each agent considering if they own some kind of private mobility mode [19] or if in their living place there is public transportation available [20], *mobility_mode* commuting mobility mode chosen in each iteration of the housing search, *time_main_activity*: commuting time in minutes, *distance_main_activity*: commuting distance depending on the housing option being considered and *commuting_cost*: cost based on the distance and the mobility mode chosen.

3.2 Process Overview

The dynamics of this model are ruled by the behaviour of ‘people agents’. Each of them is assigned a nonresidential building in the finer grained area as their workplace. An iterative process will be performed in order to find the housing option and mobility mode that best meets their needs.

Iteration 0: The initialisation process takes place in three different steps: (i) a random residential unit is assigned to each agent, (ii) based on the area where this residential unit is located and the probability of owning private means of transportation, possible mobility modes are defined (iii) travelling options will be weighed based on each profile’s criteria and a final score for each mode will be calculated as a linear combination of the importance given to each of these factors. The option that maximizes the score will be the chosen mobility mode. Criteria related to transportation preferences include both quantitative and qualitative factors such as price, resulting commuting time, the difficulty of usage and the social pattern [11]. Table 1 displays some synthetic values of this criteria for three medium income profiles [11] (modified from working status profiles into income profiles), whereas Table 2 shows the features of some of the available transportation modes [11, 20].

Table 1. Synthetic weighing values given to transportation criteria [11] according to different income profiles. These values are normalized between -1 and 0 for price, time and difficulty and between 0 and 1 for pattern importance. They will determine each ‘people agent’s’ preference towards a particular mobility choice.

	Price [-]	Time [-]	Difficulty [-]	Pattern [-]
\$45,000–\$59,999	-0.8	-0.8	-0.7	0.7
\$60,000–\$99,999	-0.7	-0.85	-0.75	0.8
\$100,000–\$124,999	-0.6	-0.9	-0.8	0.95

Note that in order to be able to make use of public transportation, both the surroundings of the residential unit and the workplace have to offer that particular commuting option. So as to calculate the time needed to commute for each mode, both the waiting time (when non-private modes are considered) and the commute itself – following the topology of the transportation network – have been taken into account.

Table 2. Price, speed and social pattern features of some of the available mobility modes [11]. Price/km and mean speed values are based on the massachusetts bay transportation authority data [20], whereas the pattern weight is a synthetic normalized weighing value between 0 and 1. A linear combination between these values and the mobility choice criteria for each income profile is performed so as to select the most suitable transportation option.

	Price/km [\$]	Mean speed [km/h]	Pattern [-]
Bike	0.01	5	0.5
Bus	0.1	20	0.4
Car	0.32	30	1

Subsequent Iterations: Once the initialisation process has been held, the procedure will change into: (i) an alternative housing option is randomly assigned to each agent (ii) this new housing option is compared to the current unit and the suitability of moving to the new alternative is evaluated based on certain factors according to each income profile (iii) the ‘time’ parameter is calculated as the resulting commuting time required using the most suitable mobility option for each housing unit as described for iteration 0. ‘People agents’ will decide to move in iteration ‘i’ if the score of the alternative housing unit is greater than the current one. Factors considered when assessing housing include quantitative data such as price or commuting time using the most suitable available means of transportation and qualitative factors like the zone preference, importance given to the unit being in this zone and diversity acceptance [9].

Diversity is calculated using the Shannon-Weaver formula following the diversity calculation methodology deployed on the CityScope platform [2]. This measurement quantitatively measures the amount of species (different income profiles in our research) in an ecosystem (the spatial unity being considered in each case) [2]. This diversity metrics will be applied either to a building, if the housing unit being considered is within the finer grained area, or to a census block group should the dwelling be located on the outskirts. This iterative process will be applied until the number of people moving in each iteration asymptotically approaches zero. Once this situation is reached and every ‘people agent’ is assigned the most convenient mobility and housing option, further mobility studies can be performed departing from this $t = 0$ situation.

Table 3. Price, diversity acceptance and zone criteria according to different income profiles (synthetic data based example). Values are normalized between -1 and 0 for price, between -1 and 1 for diversity acceptance and between 0 and 1 for zone weight. These criteria, along with the transportation criteria will point out the housing and mobility choice that maximizes the score obtained through a linear combination.

	Price [-]	Diversity acceptance [-]	Zone weight [-]
\$45,000–\$59,999	-0.8	0	0.4
\$60,000–\$99,999	-0.6	-0.3	0.5
\$100,000–\$124,999	-0.5	-0.5	0.6

3.3 Initialization and Input Data

The model initialization relies on two main types of files: the ones containing geographical information and the ones containing information regarding agents and mobility modes. The first group of files includes (1) a shapefile that consists of the block groups of the area in question, (2) a shapefile where the availability of apartments according to each block group is displayed, (3) a shapefile where the buildings of the finer grained area are incorporated, (4) a shapefile that is comprised of the public transportation system and (5) a shapefile with the road network of the region. The second type of input data consists of .csv files where (1) income profiles are defined and their characteristics such as the proportion within the population, the probability of owning a car and a bike are listed [19], (2) ‘people’ profiles define the weight given to each of the criterion when choosing a house – Table 3–, (3) these same profiles describe the criteria regarding mobility mode preferences – Table 1– and (4) different mobility modes are listed and their characteristics detailed – Table 2 –, including the price and time per distance unit, the waiting time, the difficulty, and the social pattern.

4 Model Calibration and Validation: Use Case Kendall Square

The model described above in this paper presents a generic framework where, as long as the input data is modified, the methodology can be applied to any city. In order to develop a first version of the model, Kendall Square has been analysed. Kendall Square is a neighbourhood in Cambridge, Massachusetts, – Fig. 2 – that has been transformed from an industrial area into a leading biotech and IT company hub in the last decades. As the number of companies clustering around this site has soared, the cost of housing has exploded, making it increasingly difficult for low and medium income members to live within a walking distance from their workplace [21]. The commuting patterns derived from this situation, which include high private vehicle density and saturation of the public transportation infrastructure, lead to serious mobility challenges. This section is focused on the description of the calibration process of ‘people agents’ behavioural criteria, so

that their reactions to potential housing incentives can be characterized and the consequences of their acts monitored. Input files (2) and (3) gather the aforementioned criteria and will be, therefore, inferred from this validation process, whereas file (1) is deduced from census data [19] and file (4) is an adaptation of the Massachusetts Bay Transportation Authority information [20].

The calibration process is based on the definition of two types of errors: one related to housing choices and the second one related to mobility mode patterns. Batch experiments are then deployed so as to find the combination of criteria that leads to the minimum value of the sum of both of these errors. There are eight different factors that affect the decision-making process of ‘people agents’ – housing price, diversity acceptance, preferred zone and weight given to zone [9]; transportation price, time importance, commuting difficulty and social pattern [11] – which applied to eight different income profiles, leads to a total of sixty four variables handled by batch experiments.

The housing error is defined as the difference in percentage of the distribution of each income profile in each census block group where citizens working in Kendall have certain presence. The real percentage value inferred from transportation data – the census block group where different Kendall workers live can be identified and the income profile of these approximated to the mean annual income based on census data [19] – is then compared to the resulting simulated value and the root-mean-square deviation calculated – Eq. 1 –.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2} \quad (1)$$

Y_i = real percentage of people of type i living in the neighbourhood in question

\hat{Y}_i = percentage of ‘people agents’ of type i living in that same neighbourhood as a result of the housing selection methodology

As for the mobility mode error, the resulting simulated percentages of pedestrians and car, bike, bus and T usage are compared to the Parking and Transportation Demand Management Data in the City of Cambridge [22]. This error is equally calculated as the root-mean-square deviation, where Y_i represents the percentage of people that choose a certain mobility mode i, whereas \hat{Y}_i stands for the percentage of ‘people agents’ that choose that same mobility mode once the iterative process has been performed.

As far as the exploration method is concerned, the hill climbing algorithm [23] has been applied and batch experiments have been performed, getting a reasonably good solution. The most suitable criteria combination has led to a mean housing error of 3.87% and a mean mobility error of 2.30% – Fig. 3.

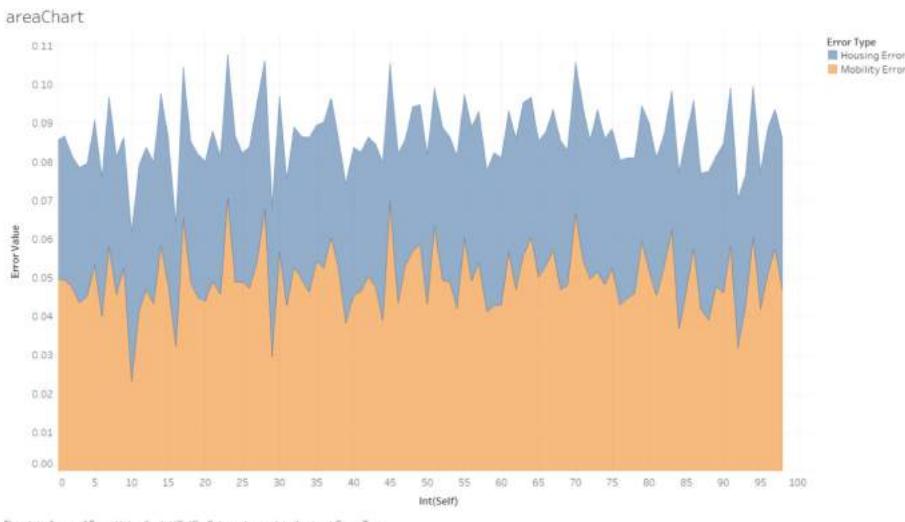


Fig. 3. Evolution of housing and mobility errors [%] in a hundred of the calibration process batch experiments.

5 Discussion

In the described model citizens' housing and mobility preferences are scored taking into account various parameters, which include price (for both housing and commuting), diversity acceptance, zone preference, time, difficulty of usage and social pattern. These parameters have been defined based on bibliography, considering that they should constitute a reduced set of variables that heavily affect the decision-making process. Should a finer-grained study be held, it should be noted that the zone preference criterion encompasses itself numerous possible attributes. The effect of greenery, high quality education centers or fresh food availability, to mention some, could also be included if a more detailed study is to be held. For a first version model, though, it has been assumed that the suggested approach was already complex enough. This same complexity, along with the metropolitan area scope, makes it challenging for the impacts resulting from changes in urban configurations to be calculated-real time, which represents a constraint when it comes to computational cost. Finally, it should be noted that in order to calibrate the criteria in question, the exploration of the design space has been performed using the hill climbing algorithm. The minimization process has resulted in acceptable housing and mobility errors. However, this algorithm heavily relies on the selected starting point and, thus, alternative algorithms could be deployed if this limitation is to be avoided in future calibration processes.

6 Conclusion

This paper presents a generic methodology where citizens' criteria towards housing and mobility choices can be tested, as well as the resulting behavioural patterns monitored. Criteria regarding mobility and residential preferences have been calibrated and validated for the particular use case of Kendall Square, where major mobility issues have arisen as a consequence of the technological development of the surroundings and the explosion of housing prices. However, variations in input data and research into combinations of criteria that are particularised for a certain city are possible following the guidelines given earlier in the document. As part of the CityScope platform, this model enables the prediction of citizens' response to housing-related urban disruptions that aim to promote the pro-social development of cities. The usage of Agent-Based Modelling opens the door to the monitorisation of people's reactions to dynamically reconfigurable zoning policies. However, the deployment of such a data-driven platform requires of real-time feedback, so that different stakeholders can benefit from the instant impact analysis of the suggested actions. Further research regarding the conversion of this model into a real-time analysis tool is being held. This new approach will need to embrace the complexity of urban dynamics, while constituting an agile and dynamic decision-making tool.

References

1. Sjoberg, G.: The origin and evolution of cities. In: *Scientific American*, vol. 213, No. 3. Scientific American, a division of Nature America, Inc (1965)
2. Alonso, L., et al.: CityScope: a data-driven interactive simulation tool for Urban design. Use Case Volpe. In: Morales A., Gershenson, C., Braha, D., Minai, A., Bar-Yam, Y. (eds.) *Unifying Themes in Complex Systems IX. ICCS 2018. Springer Proceedings in Complexity*. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96661-8_27
3. United Nations. Sustainable Development Goals (2020). <https://www.un.org/sustainabledevelopment/cities/>
4. Grignard, A., Macià, N., Alonso Pastor, L., Noyman, A., Zhang, Y., Larson, K.: Cityscope Andorra: a multi-level interactive and tangible agent-based visualization. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pp. 1939–1940 (2018)
5. Rashidi, T., Auld, J., Mohammadian, A.: A behavioral housing search model: two-stage hazard-based and multinomial logit approach to choice-set formation and location selection. In: *Transportation Research Part A 46*, pp. 1097–1107, Elsevier Ltd (2012)
6. Zinas, B., Mohd Jusan, M.: Housing choice and preference: theory and measurement. In: 1st National Conference on Environment-Behaviour Studies. Faculty of Architecture, Planning & Surveying, Universiti Teknologi MARA, Shah Ala, Selangor, Maysia, 14–15 November 2009, pp. 282–292 (2012)
7. Kim, J., Pagliara, F., Preston, J.: The intention to move and residential location choice behaviour. *Urban Stud.* **42**(9), 1621–1636 (2005)

8. Aguilera, A., Ugalde, E.: A spatially extended model for residential segregation. In: Discrete Dynamics in Nature and Society, vol. 2007, Article ID 48589. Hindawi Publishing Corporation (2007)
9. Jordan, R., Birkin, M., Evans, A.: Agent-based modelling of residential mobility, housing choice and regeneration. In: Heppenstall, A., Crooks, A., See, L., Batty, M. (eds.) Agent-Based Models of Geographical Systems, pp. 511–524. Springer, Dordrecht (2012). https://doi.org/10.1007/978-90-481-8927-4_25
10. Doorley, R., Noyman, A., Sakai, Y., Larson, K.: What's your MoCho? Real-time mode choice prediction using discrete choice models and a HCL platform. In: Urb-comp 2019, 5 August 2019, Anchorage, AK (2019)
11. Grignard, A., et al.: The impact of new mobility modes on a city: a generic approach using ABM. In: Morales, A.J., Gershenson, C., Braha, D., Minai, A.A., Bar-Yam, Y. (eds.) Unifying Themes in Complex Systems IX. ICCS 2018. SPC, pp. 272–280. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-96661-8_29
12. Lerman, S.: Location, housing, automobile ownership, and mode to work: a joint choice model. In: Transportation Research Record, pp. 6–11. Transportation Research Board (1977)
13. Clark, W., Onaka, J.: An empirical test of a joint model of residential mobility and housing choice. Environ. Plan. A Econ. Space **17**(7), 915–930 (1985)
14. Schelling, T.: Models of segregation. In: The American Economic Review, vol. 59, No. 2, pp. 488–493. Papers and Proceeding of the Eighty-first Annual Meeting of the American Economic Association (1969)
15. Grignard, A., Taillandier, P., Gaudou, B., Vo, D.A., Huynh, N.Q., Drogoul, A.: GAMA 1.6: advancing the art of complex agent-based modeling and simulation. In: Boella, G., Elkind, E., Svarimuthu, B.T.R., Dignum, F., Purvis, M.K. (eds.) PRIMA 2013. LNCS (LNAI), vol. 8291, pp. 117–131. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-44927-7_9
16. Taillandier, P., et al.: Building, composing and experimenting complex spatial models with the GAMA platform. GeoInformatica **23**(2), 299–322 (2019)
17. United States Government's Open Data (2020). <https://www.data.gov/>
18. PadMapper. Apartments for Rent from the Trusted Apartment Finder (2020). <https://www.padmapper.com/>
19. United States Census Bureau. Census Profiles (2020). <https://data.census.gov/cedsci/>
20. MBTA. Massachusetts Bay Transportation Authority (2020). <https://www.mbta.com/>
21. Sisson, P.: As top innovation hub expands, can straining local infrastructure keep pace? (2018). <https://www.curbed.com/2018/11/6/18067326/boston-real-estate-cambridge-mit-biotech-kendall-square>
22. Parking and Transportation Demand Management Data in the City of Cambridge (2014). <https://www.cambridgema.gov/CDD/Transportation/fordevelopers/ptdm>
23. Russell, J., Norvig, P.: Artificial Intelligence. A Modern Approach. Prentice-Hall, USA (1995)



Reasoning in the Presence of Silence in Testimonies: A Logical Approach

Alfonso Garcés-Báez^(✉) and Aurelio López-López

Computational Sciences Department, Instituto Nacional de Astrofísica,
Óptica y Electrónica, Sta. Ma. Tonantzintla, Puebla, Mexico
{agarces,allopez}@ccc.inaoep.mx

Abstract. An implicature, as opposed to a logical or formal implication, allows to make linguistic inferences during a conversation. The omission of an answer (silence) also allows to make linguistic inferences in some contexts, according to Swanson. Once previous studies of silence and the conversational implicature of Grice are reviewed, we propose definitions and semantics of three kinds of silence, when knowledge is represented in logic. As a case study for the semantics, we chose a puzzle formulated by Wylie and then apply answer set programming to generate models that reveal new information and different possibilities, including non-monotonicity in knowledge bases, that lie behind the omission in the context of testimonies. Also, a connection between false statements and silence emerged.

Keywords: Silence · Omission · Testimonies · Knowledge representation · Reasoning · Logic

1 Introduction

Both silence and omission have been studied from different points of view. For instance, from the point of view of semiotics, Umberto Eco [4] stated that silence is a sign and proposes a *table of possibilities* between speaker and the listener, where the transmission of silence can be voluntary or involuntary. According to [1], the variants for the interpretation of silence are *anonymous* and *not anonymous*, where the second can occur as: with identification or face-to-face. In the same work, silence is also examined in the communication process, in terms of prisoner's dilemma.

Dyne et al. [3] propose the *defensive*, *acquiescent*, and *pro-social* silence occurring in organizations. Kurzon defines silence by language in [12], focusing mainly on three types of silence: *psychological*, *interactive*, and *socio-cultural*. In data communication, there are three kinds of acoustic silence in conversation: *pauses*, *gaps*, and *lapses*. Pauses refer to silences within turns, gaps refer to short silences between turns, and lapses are longer or extended silences between turns [2].

Context is a key element for the interpretation of silence. For instance, in court, during an interrogation, silence is interpreted to the detriment of whom decides to remain silent. This is immediately understood as the person hides something. Other example is when we have the interpretation of silence by police officers with suspects that emit the phrase *no comment* instead [15].

Intentional silence can also signal loyalty to a group. According to [12], if we want to interpret intentional silence, first we have to discard the modal *can* that expresses unintentionality, as in *I can not speak*. Afterward, four manners remain to interpret. First, *I may* not tell you; second, *I must* not tell you; third, *I shall* not tell you, and finally, *I will* not tell you. First two manners are considered external intentional silences *by order*, while last two are internal *by will*.

From several of these above mentioned studies on silence [3, 4, 12, 15], we proposed previously a classification of the uses of silence that can be found in [6].

An *ommissive implicature*, as stated by Swanson [17], asserts that in certain contexts, not saying *p* generates a conversational implicature: that the speaker didn't have sufficient reason, all things considered, to say *p*.

A formalization of intentional silence in terms of logic has not been done, as far as we know. However, [11] attempts an *informal logic*. In [5], a first contribution in this direction is done.

This paper proposes a logical conceptualization to the study of omission of answers in a testimonial context, by focusing on a puzzle formally expressed to analyze the implications of three interpretations of omission represented, through the predicate Says (previously employed in [8] and [10]).

We solve as case study two versions of a puzzle, generating models with Clingo in one of them, to delineate the relationship that could exist between possible false statements and the interpretation of omission as intentional silence, in this testimonial context.

The organization of the paper is: Sect. 2 contains the concept of conversational implicature and some formal definitions, in Sect. 3 we describe the strategy we use and the puzzle taken as case study. Section 4 includes three interpretations, some combinations of them, and their consequences for the case study. Section 5 concludes and details work in progress.

2 Background

We discuss two main concepts for our logical approach, conversational implicature and answer set programming.

2.1 Grice's Sealed Lips

Grice [10] explains conversational implicature as a potential inference that is not a logical implication, and that is closely connected with the meaning of the word *says*.

The Cooperative Principle (CP) is the basis to explain implicature, and consists of participants making their contribution in a conversation, as required in the scenario in which occurs, for the accepted purpose or direction of their speech exchange. The CP has four categories that include the maxims of Grice:

1. Quantity. Make your informative contribution as required for current exchange purposes. Do not make more informative contributions than required.
2. Quality. Do not say what you think is false. Do not say that for which you lack adequate evidence.
3. Relation. Be relevant.
4. Manner.

Category 4 is related to the way it is said and considers a supermaxim ‘Be perspicuous’, along several maxims such as:

- Avoid obscurity of expression.
- Avoid ambiguity.
- Be brief (avoid unnecessary prolixity).
- Be orderly.

As pointed out by Grice, a talk exchange participant may fail to fulfill a maxim and the CP; i.e. he may say, indicate, or allow it to become plain that he is unwilling to cooperate according to the maxim. He may say, for instance, *I cannot say more; my lips are sealed*, and with this, breaking the communication. In contrast, we consider that in some contexts, silence could be interpreted without breaking communication.

The representation of saying and not saying is important in our semantics, which is the reason for introducing the following formal definition of this predicate.

Definition 1. *Says(A, P, T/F)* denotes that the agent A says that P is true (T) or false (F).

2.2 Answer Set Programming

While the antecedents of Prolog are in automatic theorem provers, those of Answer Set Programming (ASP) are in deductive databases and satisfiability testing. For ASP, starting from a representation of the problem, a solution is given by a model of the representation.

ASP paradigm is utilised in our research to explore the implications of silence in our case study, since is closely related to intuitionist logic, i.e. these are based on the concept of *proof* rather than *truth* (as previously shown in [9] for intuitionist logic). This is a way of doing logical programming by computing stable models for the problems at hand [8]. A stable model is a belief system that holds for a rational agent. Under this approach:

- Solutions go beyond answering queries.

- The solution of computational problems is done by reducing them to finding answer sets of programs.
- In principle, any problem in NP-complete class can be solved with ASP without disjunction.
- More-complex problems can be solved when disjunction is included.

3 Case Study

Our aim is to show some possibilities that open when intentional silence or omission occurs during testimonies, as a computational resource that helps for decision-making.

We start from a puzzle with three suspects under the assumption that one tells the truth, another asserts half truth, and the last one lies. With the purpose of uncovering the consequences of silence, we model this puzzle under a new assumption: everyone tells the truth except the criminal, and we apply our semantic rules to reveal some relationships between silence and falsehood.

3.1 Method

In mathematics, there is a wide variety of proof techniques to prove that $A \rightarrow B$ such as Direct, Indirect, Contradiction, Contrapositive, Construction, Induction, and so on [16]. As part of that list, there is the Choose Technique that works forward from A and the fact that the object has certain property (see Fig. 1). This also works backward from the something that happens.

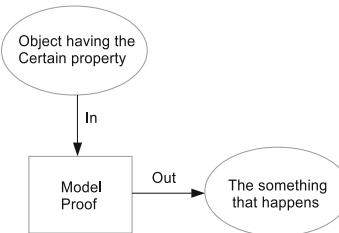


Fig. 1. A model proof for the Choose technique.

We employ logic puzzles since the solutions to some of these problems could have practical implications, and in some way they synthesize actual situations.

Figure 2 shows the preliminary considerations (*preconditions*) that consist of selecting a puzzle with testimonials and implementing (programming) it to reach a contextualized solution, employing common sense rules. In the *process*, the types of omission are chosen to generate models, that are then analyzed to try to answer some *key questions*.

PRECONDITIONS	PROCESS	KEY QUESTIONS
<ul style="list-style-type: none"> Selection of puzzle with testimonies Formalization of testimonies and common sense rules according to context Problem solving with answer set programming 	<ol style="list-style-type: none"> Omission type selection Omission modeling and its variants Analysis and feedback 	<ul style="list-style-type: none"> All cases have a solution? Are there models that contain the answer to the original problem? What is the consequence of the silence appearing?

Fig. 2. Strategy.

An implementation of ASP is Clingo [7] that allows to find, if there exists, the answer set (stable model) of a logical program. For our case study, Clingo is employed to generate answer sets for the problem, and in particular explore the implications of the interpretations formulated. Additionally, Python serves to update the logical program [18].

3.2 Riddle the Criminal

There is a puzzle that includes testimonies of several people (taken from [19]), and allows to model and examine the interpretations of silence. The riddle is phrased as:

Three men were once arrested for a crime which beyond a shadow of a doubt had been committed by one of them. Preliminary questioning disclosed the curious fact that one of the suspects was a highly respected judge, one just an average citizen, and one a notorious crook. In what follows they will be referred to as Brown, Jones, and Smith, though not necessarily respectively. Each man made two statements to the police, which were in effect:

Brown:

- b₁: I didn't do it.*
- b₂: Jones didn't do it.*

Jones:

- j₁: Brown didn't do it.*
- j₂: Smith did it.*

Smith:

- s₁: I didn't do it.*
- s₂: Brown did it.*

The original completion is:

Further investigation showed, as might have been expected, that both statements made by the judge were true, both statements made by the criminal were false, and of the two statements made by the average man one was true and one was false. Which of the three men was the judge, the average citizen, the crook?

However, the puzzle can be formulated more naturally as follows:

Every person involved tells the truth except, possibly, the criminal. Who committed the crime?

To analyze the solutions of these two versions, the following logic-based representation of testimonies is done:

- $b_1: \text{says}(\text{brown}, \text{innocent}(\text{brown}), T).$
- $b_2: \text{says}(\text{brown}, \text{innocent}(\text{jones}), T).$
- $j_1: \text{says}(\text{jones}, \text{innocent}(\text{brown}), T).$
- $j_2: \text{says}(\text{jones}, \text{innocent}(\text{smith}), F).$
- $s_1: \text{says}(\text{smith}, \text{innocent}(\text{smith}), T).$
- $s_2: \text{says}(\text{smith}, \text{innocent}(\text{brown}), F).$

To solve the puzzle, we have to keep in mind the following conflicting statements:

1. $\{b_1, s_2\} \vdash \perp$
2. $\{j_1, s_2\} \vdash \perp$
3. $\{j_2, s_1\} \vdash \perp$

A straightforward and traditional way to solve this type of puzzle is through a matrix representation. For instance, a *SaysMatrix*, as specified in Definition 2, illustrates the speech acts involved in a context, in terms of the predicate $\text{Says}(X, Y, T/F)$.

Definition 2. A *SaysMatrix* has rows that correspond to the agent that denies (*F*) or asserts (*T*) a predicate of another agent found in a column. If there are several predicates, the intersection of columns and rows is the pair $\langle \text{Predicate}, T/F \rangle$ or empty; if the matrix is associated with only one predicate, the intersection can have only *T/F*.

Figure 3-i presents the testimony of the suspects represented with the predicate $\text{Says}(x, \text{innocent}(y), T/F)$ previously defined.

Figure 3-ii depicts the solution to the original puzzle. This result is found based on the preconditions of the original completion by trial and error. The only solution turns out that Brown is the culprit.

Our interest is to know the implications of omission of a testimony or silence in the context of the case study, that is, if by considering silence instead of false statements, we can also reach a solution to the puzzle, when analyzing the alternative (natural) completion of the case study puzzle. Under this assumption, the solution to the puzzle varies, where the criminal turns out to be Smith, as illustrated in Fig. 3-iii.

The program Criminal.lp (for Clingo, see Appendix A) implements the testimony of these three people, along with common sense knowledge. So, the program generates the solution: $\text{criminal}(\text{smith})$.

Next, we present and exemplify some definitions of silence to explore their relationship, regarding the solution of the original puzzle that serves as our case study.

<i>innocent</i>	Brown	Jones	Smith
Brown	T	T	-
Jones	T	-	F
Smith	F	-	T

i. Testimonial SaysMatrix

	Brown	Jones	Smith
Brown	F	T	-
Jones	F	-	F
Smith	T	-	T

ii. Original puzzle solution

	Brown	Jones	Smith
Brown	T	T	--
Jones	T	-	T
Smith	F	-	F

iii. Natural puzzle solution

Fig. 3. SaysMatrix and solutions of the puzzle the criminal.

4 Interpreting Omission

Starting from the representation of the puzzle, we then describe and explore two variants of defensive silence and one of acquiescent silence related to the context of interest.

The first silence to interpret is Defensive, that occurs when an agent intentionally simply decides to remain quite, mainly by fear. The second interpretation is for Acquiescent Silence, i.e. asserting with silence what others have said, commonly caused by resignation. For both cases, we examine their consequences.

4.1 Defensive Silence

This kind of silence is intentional and proactive, intended to protect the self from external threats, and is defined next.

Definition 3. *Defensive Silence. Withholding relevant ideas, information, or opinions as a form of self-protection, based on fear [3].*

To reason in the presence of this kind of silence, the following rule is stated to interpret it.

TDS semantic rule:

P_{A_i} denotes Total Defensive Silence of A_i understood as:

$$P_{A_i} = P - X_{A_i}$$

With $(1 \leq i \leq n)$, n is the number of interacting agents; P is a logic program or knowledge base, and $X_{A_i} = Says(A_i, *, *)$ is all that the agent A_i Says.

When someone faces this kind of silence of one or more of those involved in a case under investigation, he can not count on that particular information. Hence, to take into account such behaviour in practical terms, we have to remove the declaration of those people, that involves *delete says(agent, *, *)*.

So, expressing this kind of silence for the case study (puzzle); what would happen if silence with common sense is presented as a possibility? Can the interrogator or judge reach any conclusion when some of the suspects decide to intentionally remain quiet?

Considering silence, i.e. (total) defensive silence (TDS), for each person giving their testimony, through its implementation as metaprogramming in Python (see Appendix B), would imply executing:

```
t_def_silence('Criminal.lp', 'brown')
t_def_silence('Criminal.lp', 'jones')
t_def_silence('Criminal.lp', 'smith')
```

and running:

```
clingo 0 (kb-tds-agent).pl
```

We obtain those presumably culpable, i.e. as a result of the silence of one or more persons, we can examine who turns into a candidate to blame.

Table 1 includes the possible outcomes (guilty) if suspects decide intentionally to omit their testimonies. One can observe that the perpetrator can be anyone depending on who remains silent. The following comments are pertinent for the different possibilities:

1. The first case coincides with the complete puzzle, i.e. every suspect provides his statement and Smith emerges as guilty.
2. If Brown remains silent, either of the other suspects can be the perpetrator.
3. When Jones remains quiet, either of the other two suspects may be guilty.
4. Smith is guilty even if he tries to protect himself with silence.

Table 1. Total defensive silence model for silent agent

Silent agent	Presumably culprit
{}	{Smith}
Brown	{Smith, Jones}
Jones	{Smith, Brown}
Smith	{Smith}

In TDS, we dispense of all the statements of a particular witness, however it could happen that a witness decides to reveal only part of his version. This is the case of a partial defensive silence (PDS). In this scenario, we can now wonder: What part of the testimony could be convenient to silence in the case

of suspects? Who would be the culprit in the event that some person decides to remain partially silent? What possibilities would each of the suspects has if, before giving his allegation, he had access to the testimony of the others? Now, the rule to interpret this kind of silence turns into the following.

PDS semantic rule:

P_{A_i, p_j} denotes Partial Defensive Silence of A_i interpreted as:

$$P_{A_i, p_j} = P - \{Says(A_i, p_j)\}$$

With $(1 \leq i \leq n)$, n is the number of agents interacting; $(1 \leq j \leq m)$; A_i is an agent; m is the number of utterances/assertions expressed by A_i , and p_j is a particular assertion of A_i .

To explore this other scenario, we would have to first execute:

```
p_def_silence(kb,agent,predicate)
```

and then run:

```
clingo 0 (kb-pds-agent-predicate).pl
```

Table 2. Partial defensive silence model for predicate

Silent predicate	Presumably culprit
b_1	{Smith}
b_2	{Smith}
j_1	{Smith}
j_2	{Smith}
s_1	{Smith}
s_2	{Smith}

By detailing the PDS analysis, as shown in the Table 2, we can observe that no predicate alone has the decisive capability to generate another possible culprit, since there is only one model in all cases.

4.2 Acquiescent Silence

The old saying “silence is consent” is behind the third interpretation of silence, and expresses a passive disengaged attitude. This is defined as:

Definition 4. *Acquiescent Silence. Withholding relevant ideas, information, or opinions, based on resignation [3].*

Now this kind of silence is interpreted according to the rule stated next.

AS semantic rule:

P'_{A_i} denotes Acquiescent Silence of A_i understood as:

$$P'_{A_i} = P_{A_i} \cup (\{Says(A_j, *)\} \circ \lambda)$$

With $i \neq j$, $(1 \leq i, j \leq n)$, n is the number of involved agents; P_{A_i} is TDS for A_i ; $\lambda = \{A_j/A_i\}$, and the operator \circ with λ substitution expresses the exchange of A_j by A_i on the *Says* subset of agent A_i .

The way to put in practice this interpretation is by omitting the whole person's testimony and adding new rules related to what is implicitly assuming with silence. For example, in the case of Smith, the following code has to be executed:

```
acq_silence('Criminal.pl', 'smith')
```

which causes that:

1. A TDS corresponding to Smith agent is applied, that is, eliminate his $Says(x, y, z)$ predicates.
2. Add the testimonies of other witnesses as if they were his, that is, put in his mouth what others say. That is, insert $Says$ (*other-agents*, *, *) instances.

The acquiescent silence (AS) for Smith, where he again turns out to be guilty, is obtained by the execution of the program:

```
clingo 0 Criminal-as-smith.pl
```

The answer was similar to model of the problem in its natural version.

Table 3. Acquiescent silence model for silent agent

Silent agent	Presumably culprit
{}	{Smith}
Brown	<i>Unsatisfiable</i>
Jones	<i>Unsatisfiable</i>
Smith	{Smith}

The solutions found for the puzzle when a person is silenced under the AS interpretation, are listed in Table 3. Notice that for silence of Brown or Jones, no model can be obtain but again Smith is incriminated even if he recurs to AS.

4.3 False Statements or Silence

Is there a relationship between silence and false statements? As we can notice in Table 4, in the context of the case study, further information is hidden behind silence and more possibilities are opened.

Table 4. Combined silence

#	TDS	PDS	Presumably culprit
1	Brown	j_1	{Smith, Jones}
2		j_2	{Smith, Jones}
3		s_1	{Smith, Jones}
4		s_2	{Smith, Jones}
5	Jones	s_1	{Smith, Brown}
6		s_2	{Smith, Brown}
7		b_1	{Smith, Brown}
8		b_2	{Smith, Brown}
9	Smith	b_1	{Smith}
10		b_2	{Smith, Jones}
11		j_1	{Smith, Brown}
12		j_2	{Smith}

In the solution of the puzzle in its original version (as depicted in Fig. 3-ii), Jones is who declared two false statements (j_1 and j_2) and Brown only the first one (b_1), these correspond to Jones' total defensive silence and a partial defensive silence of Brown, as we can notice in row 7 of the Table 4 where, of the two models that are obtained, one is the solution to the puzzle with its original version (Brown) and the other is the solution to the puzzle in the new (natural) version (Smith).

4.4 Discussion

As the case study showed, when someone recurs to defensive silence, possibilities are opened while with acquiescent silence, they are reduced due to the increment in conditions that have to be met, even to the degree of finding no solution at all (as Table 3 showed). The characteristics of the puzzle allow to perceive, intuitively, that behind the silence there is important information, including the solution to some variant of it, as shown in previous section.

An important fact that we want to bring out is when two programs are *equivalent* with respect to the semantic answer set. For these models, a definition for *equivalence* is: two programs are equivalent if they have the same answer set [13]. The program, whose known solution is Smith, and the other programs obtained for the interpretations of silence, are equivalent from the point of view of the achieved result.

As part of the results, we reformulate automatically a knowledge base (KB) of a logical program P (speech acts, rules or actions) based on the information inherent in the omission, obtaining an equivalent program P' that has fewer rules than the original program. That is:

Let P and P' two logical programs with predicates $says(x, y, z) \in X$ and $X \subset P$. If $P' == (P - X)$ such that X is “silenced” in P , then P' is a reduction of P .

Moreover, this formulation of interaction as speech acts (represented as predicates $Says(x, y, z)$) under the assumption of silence of one or more of the interacting agents shows non monotonicity, because allows to draw tentative conclusions, in particular:

- With TDS and the silence of Smith, the first found answer led to Smith as solution (see Table 1). This requires two rules less in the knowledge base (KB_1).
- With AS and the silence of Smith, the perpetrator comes out also as Smith (see Table 3). Here, two additional rules are required, i.e. $-2 + 4(KB_2)$.

Considering the cardinality of the knowledge bases, we observe that the following holds:

$$|KB_1| \leq |KB| \leq |KB_2|.$$

5 Conclusions

Given the close relation between Answer Set Programming (ASP) and intuitionist logic, each model generated in ASP can have a proof in the logic [14]. In consequence, each of the applications of the interpretations of silence and their possible combinations, intuitively, involve a proof within the belief system constructed from the rational-agents statements.

As shown in the case study, behind silence there is valuable information that can be taken into account for decision making. So, implicatures can be achieved if intentional silent is interpreted along the appropriate context.

In our case study, we can observe that there exists a relationship between the falsehood of statements and silence (omission).

The study and modeling of silence can be useful to reach intelligent human-computer interaction. Also, this research on silence can be usable to analyse scenarios in legal cases involving testimonies. A strategy has been proposed.

Scenarios where three kinds of silence are involved, e.g. a participant is recurring to defensive silence (total or partial) and other to acquiescent silence, were explored in this study. Other possible interpretation to consider is that of prosocial silence, i.e. retaining work-related information or opinions for benefit of other people or organization.

We would like also to bring the interpretations of silence to a more general framework for agent interaction, beyond answer set programming. For instance, the different semantics can be brought in dialogue systems where virtual agents attend appropriately the omission of answers during interaction, as in dialogue games.

Acknowledgments. The first author thanks the support provided by Consejo Nacional de Ciencia y Tecnología and Consejo de Ciencia y Tecnología del Estado de Puebla. The second author was partially supported by SNI.

A Criminal.lp for Clingo 4.5.4

```
%% Puzzle 51 of Wylie [19]
%% for the natural solution.
%%
suspect(brown;jones;smith).
%
%% Brown says:
says(brown,innocent(brown),1).
says(brown,innocent(jones),1).
%
%% Jones says:
says(jones,innocent(brown),1).
says(jones,innocent(smith),0).
%
%% Smith says:
says(smith,innocent(smith),1).
says(smith,innocent(brown),0).
%%%%%%%%%%%%%
%
%% Everyone, except possibly for the criminal, is telling the truth:
holds(S) :- says(P,S,1),
           -holds(criminal(P)).
-holds(S) :- says(P,S,0),
           -holds(criminal(P)).
%
%% Normally, people aren't criminals:
-holds(criminal(P)) :- suspect(P), not holds(criminal(P)).
%
%% Criminals are not innocent:
:- holds(innocent(P)),holds(criminal(P)).
%
%% For display:
criminal(P) :- holds(criminal(P)).
%
%% The criminal is either Brown, Jones or Smith, (exclusively):
holds(criminal(brown)) | holds(criminal(jones)) | holds(criminal(smith)).
#show criminal/1.
```

B Silence.py Prototype for Python 3.7

```
# Definition of Total Defensive Silence
# Input: knowledge base or logic program, and agent to silence
# Output: new knowledge base or logic program named 'kb'-'tds'-'agent'.lp
def t_def_silence(kb,agent):
    f=open(kb,'r')
    g=open(kb[0:len(kb)-3]+'-'+tds+'+'agent+'.lp','w')
    for line in f:
        if 'says('+agent == line[0:5+len(agent)]:
            line='%' +line
            g.write(str(line))
    f.close()
    g.close()
```

```

# Definition of Partial Defensive Silence
# Input: knowledge base or logic program, and agent to silence
# Output: new knowledge base or logic program named 'kb'-'pds'-'agent'-'predicate'.lp
def p_def_silence(kb,agent,predicate):
    f=open(kb,'r')
    g=open(kb[0:len(kb)-3]+'-'+'pds'-'agent'-''+predicate+'.lp','w')
    for line in f:
        if 'says('+'agent'+','+'predicate'+')' == line[0:5+len(agent)+len(predicate)+2]:
            line='%' +line
            g.write(str(line))
    f.close()
    g.close()

# Definition of Acquiescent Silence
# Input: knowledge base or logic program, and agent to silence
# Output: new knowledge base or logic program named 'kb'-'as'-'agent'.lp
def acq_silence(kb,agent):
    f=open(kb,'r')
    g=open(kb[0:len(kb)-3]+'-'+'as'-''+agent+'.lp','w')
    for line in f:
        if 'says('+'agent' == line[0:5+len(agent)]:
            line='%' +line
        elif 'says(' == line[0:5]:
            i=line.index(',')
            line_new='says('+'agent'+line[i:len(line)]
            g.write(str(line_new))
            g.write(str(line))
    f.close()
    g.close()

```

References

- Bohnet, I., Frey, B.S.: The sound of silence in prisoner's dilemma and dictator games. *J. Econ. Behav. Organ.* **38**(1), 43–57 (1999)
- Chatwin, J., Bee, P., Macfarlane, G.J., Lovell, K., et al.: Observations on silence in telephone delivered cognitive behavioural therapy (T-CBT). *J. Linguist. Prof. Pract.* **11**(1), 1–22 (2014)
- Van Dyne, L., Ang, S., Botero, I.C.: Conceptualizing employee silence and employee voice as multidimensional constructs. *J. Manag. Stud.* **40**(6), 1359–1392 (2003)
- Eco, U.: Signo. Labor, Barcelona (1994)
- Garcés-Báez, A., López-López, A.: First approach to semantics of silence in testimonies. In: International Conference of the Italian Association for Artificial Intelligence, pp. 73–86. Springer (2019)
- Garcés-Báez, A., López-López, A.: Towards a semantic of intentional silence in omisive implicature. *Digitale Welt* **4**(1), 67–73 (2020)
- Gebser, M., Kaminski, R., Kaufmann, B., Ostrowski, M., Schaub, T., Wanko, P.: Theory solving made easy with clingo 5. In: OASIcs-OpenAccess Series in Informatics, vol. 52. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2016)
- Gelfond, M., Kahl, Y.: Knowledge Representation, Reasoning, and the Design of Intelligent Agents: The Answer-Set Programming Approach. Cambridge University Press, Cambridge (2014)
- Gödel, K., Anzeiger Akademie der Zum intuitionistischen Aussagenkalkül: Wissenschaften wien, math.-naturwissensch. Klasse **69**, 65–66 (1932)

10. Grice, H.P.: Logic and conversation. In: Cole, P., et al. (eds.) *Syntax and Semantics: Speech Acts*, vol. 3, pp. 41–58 (1975)
11. Khatchadourian, H.: *How to Do Things with Silence*, vol. 63. Walter de Gruyter GmbH & Co KG (2015)
12. Kurzon, D.: The right of silence: a socio-pragmatic model of interpretation. *J. Pragmat.* **23**(1), 55–69 (1995)
13. Lifschitz, V., Pearce, D., Valverde, A.: Strongly equivalent logic programs. *ACM Trans. Comput. Log. (TOCL)* **2**(4), 526–541 (2001)
14. Pearce, D.: Stable inference as intuitionistic validity. *J. Log. Program.* **38**(1), 79–91 (1999)
15. Schröter, M., Taylor, C.: *Exploring Silence and Absence in Discourse: Empirical Approaches*. Springer, Heidelberg (2017)
16. Daniel, S.: *How to Read and Do Proofs: An Introduction to Mathematical Thought Processes*. Wiley, Hoboken (2005)
17. Swanson, E.: Omissive implicature. *Philos. Top.* **45**(2), 117–138 (2017)
18. VanderPlas, J.: *Python Data Science Handbook: Essential Tools for Working with Data*. O'Reilly Media Inc., Sebastopol (2016)
19. Wylie, C.R.: *101 Puzzles in Thought and Logic*, vol. 367. Courier Corporation (1957)



A Genetic Algorithm Based Approach for Satellite Autonomy

Sidhdharth Sikka^{1,2}(✉) and Harshvardhan Sikka^{1,2}

¹ Manifold Computing, Atlanta, Georgia

sidhsikka1998@g.ucla.edu, harsh@manifoldcomputing.com

² Georgia Institute of Technology, Manifold Computing, OpenMined,
Atlanta, Georgia

Abstract. Autonomous spacecraft maneuver planning using an evolutionary computing approach is investigated. Simulated satellites were placed into four different initial orbits. Each was allowed a string of thirty delta-v impulse maneuvers in six cartesian directions, the positive and negative x, y and z directions. The goal of the spacecraft maneuver string was to, starting from some non-polar starting orbit, place the spacecraft into a polar, low eccentricity orbit. A genetic algorithm was implemented, using a mating, fitness, mutation and crossover scheme for impulse strings. The genetic algorithm was successfully able to produce this result for all the starting orbits. Performance and future work is also discussed.

Keywords: Evolutionary computation · Satellite · Autonomous systems

1 Introduction

Applications of orbital technology grow as the cost of producing satellites and launching them decrease. For the first time in history, diminished launch and development costs are making it possible to create distributed satellite systems. These systems have numerous operational advantages over monolithic satellite design, including but not limited to robustness to the consequences of failure in individual satellites, decreased cost of implementation and replacement due to the size of the units, and being in multiple locations in space. Increases in computing power and miniaturization of the associated platforms has also increased the computational resources available on-board small satellites. This allows for the use of distributed satellite systems in previously intractable problem domains, including planet-wide monitoring for tasks like gathering climate data and tracking wildlife migratory patterns [2]. Additionally, these advances have also opened up possibilities for coordination and autonomy in distributed satellite systems. If a group of satellites were to coordinate, they could accomplish objectives like flying in formation, assembling into more complex structures, retrieval and repair, etc. A short history of the prior research and application

of satellite autonomy and distributed coordination (Satellite ADC) is presented below.

1.1 History of Autonomy and Distributed Space Systems

Autonomy in single satellites and spacecraft has been implemented in older missions than autonomy among distributed systems of spacecraft. Implementations of autonomous spacecraft date back to 1997, when the Mars Pathfinder mission, in particular the rover *Sojourner*, employed simple autonomy features including simple landmark based navigation. A year later, *Deep Space 1* was performing autonomous task allocation [3] and image based navigation [4], and three years later in 2000, NASA's *Earth Observing 1* was autonomously imaging areas of interest on the Earth [5]. The first major mission successfully demonstrating a form of distributed autonomy was the NASA NODES mission. This mission deployed from the ISS in 2016 and was a demonstration of task allocation among multiple satellites using a coordination scheme called negotiation. Negotiation is when satellites establish a connection with each other, then allocate tasks among each other based on orbital parameters, telemetry and resource information [6]. The mission was successful in accomplishing its objectives. Currently, NASA has introduced a distributed autonomous mission paradigm called Autonomous Nano-Technology Swarm (ANTS) and have several missions in the planning stage following this paradigm including the Saturn Autonomous Ring Array (SARA), a swarm of small satellites that will closely observe the rings of Saturn [2].

1.2 Related Work

Though the implementations of distributed, autonomous space systems have been few and far in between, exploration of the problem has been a popular topic of academic study around the world. This research has broken down the general problem of satellite autonomy and coordination into several sub-problems, including but not limited to retrieval and repair scenarios, satellite tasking, earth observation, space observation, formation flight and science scenarios [2]. Use of indirect coordination methods, like Ant Colony Optimization algorithms or stigmergy methods, and direct coordination like in the negotiation scheme used in the NODES mission, have been explored for spacecraft coordination [6]. A common thread, regardless of the coordination plan, is to use learning or optimization to improve coordination over time. Among the research into spacecraft autonomy, which is sometimes separate from spacecraft coordination, several schema have been proposed to separate the various necessary on-board processing into parts. Most of these schema separate on-board processing into a reactive layer, which takes sensor input and reacts in low level ways, and a reflective layer, which takes information from the reactive layer and self reflects on performance. This reflective layer adjusts the reactive layer to achieve better performance towards high level goals. Across the literature, this seems to be the simplest autonomy schema that others are more complicated versions of.

1.3 Advantages of Learned Approaches

The work presented in this paper is part of a larger effort to approach the problem of Satellite ADC with simulation and optimization, rather than through the development of robust (meaning error resistant in this context) deterministic algorithms. Advances in machine learning and simulation over the last 20 years mean that the space of possible pathways to satellite autonomy are more expansive than they were during the development of *Deep Space 1*. It is now possible for a satellite to carry out numerous simulations onboard to test potential actions it may take. The development of learning methods based on achieving a goal, including reward based reinforcement learning or fitness based evolutionary algorithms, allow for a learned approach based on simulated or real outcomes as compared to traditional supervised learning based approaches that require a training dataset representative of the problem domain [1]. This is ideal for autonomy, which is defined by independent accomplishment of goals. So at this time, the alignment between Satellite ADC and learning methods certainly is present. The advantages of this approach, as opposed to developing robust algorithms to handle tasks autonomously, are manifold. Robust algorithms are developed by humans, and don't flexibly adapt to the problem domain. Learning from simulation enables automatic accommodation of circumstances which are not foreseen by a human developer, but do occur within simulation. Then as simulation becomes more advanced, and the problem domain more complex, it becomes more important that control algorithms learn from simulation. Robust algorithms also do not generalize, they are task specific, whereas learned approaches do generalize and can be adjusted to any task provided the reward or fitness functions are updated. Due to the reasons enumerated above, general application of learning systems, through the use of various simulation and optimization methods, promises significant impact in the problem of Satellite ADC.

1.4 Evolutionary Algorithm Preliminaries

An evolutionary algorithm is a type of optimization algorithm that is inspired by biological evolution. Rather than “learn” a good solution to a given problem through a system of well defined rewards and adjustments as is the case in reinforcement learning approaches, an evolutionary algorithm creates a population of candidate. These solutions “mate” with each other in some way to produce more solutions. This mating method must preserve some important aspects of the parent solutions, while also allowing the offspring solutions to differ from their parents to adequately search the solution space. Solutions have some fitness defined by the goal of the algorithm. In order for better solutions to be born, solutions with greater fitness must mate with candidates with lower fitness. This can be accomplished with either strong selection, where the less fit don't get to mate at all, or weak selection, where the less fit simply have a lower probability or proportion of mating. This is the basic form of an evolutionary algorithm, there has been significant work investigating the use of additional algorithmic steps and processes to prevent population stagnation and improve convergence

[1]. We outline some of these extensions to basic evolutionary algorithms in the experimental descriptions below.

In the following sections, we will elaborate on the goal and setup for this experiment, as well as the details of the algorithm that was developed for this experiment.

2 Genetic Algorithm Implementation

2.1 Goal Task Description

The goal was for a simulated spacecraft, placed in some starting orbit, to perform a series of thirty delta-v impulse maneuvers, or instantaneous changes in velocity, into a polar, circular orbit. The number 30 for the size of the series was chosen as a baseline, and the number of delta-v maneuvers allowed would realistically be determined based on spacecraft parameters. These parameters may include factors like onboard fuel, orbital position, and target orbit. The goal of the program was to implement a genetic algorithm which, not provided with any special start series of maneuvers, could produce a series of maneuvers which resulted in a highly polar and highly circular final orbit. Out of the six Keplerian orbit elements, the only ones considered in this experiment were eccentricity and inclination.

2.2 Experimental Setup

The experiment was carried out using the Poliastro Python library, an open source orbital dynamics library. Four simulated spacecraft were placed into different, non polar and non circular orbits, as can be seen in Fig. 1. The size of the delta-v maneuvers they could do was constrained, and the directions were constrained to the six Cartesian directions, positive and negative x, y and z. They would perform one delta-v maneuver every 20 min, between maneuvers their orbit was propagated 20 min. At every 20 min interval, they could also remain idle instead of performing a maneuver.

2.3 Basic Algorithm

The basic form of the implementation of this genetic algorithm included four parts. Devising a fitness score, generating a population of solutions, mating these solutions, and populating a new generation with their children. The fitness score for this experiment was composed of two factors, the eccentricity and the inclination. The inclination of a polar orbit is 90° , so the fitness score was the distance of the inclination of the final orbit from 90° summed with the eccentricity of the final orbit. The final orbit in this case refers to the resultant orbit of a particular maneuver sequence. The lower the fitness score, the more fit the maneuver sequence. The population generation was performed by using a random number generator to generate a random number between 0 and 6 and creating a 30 digit

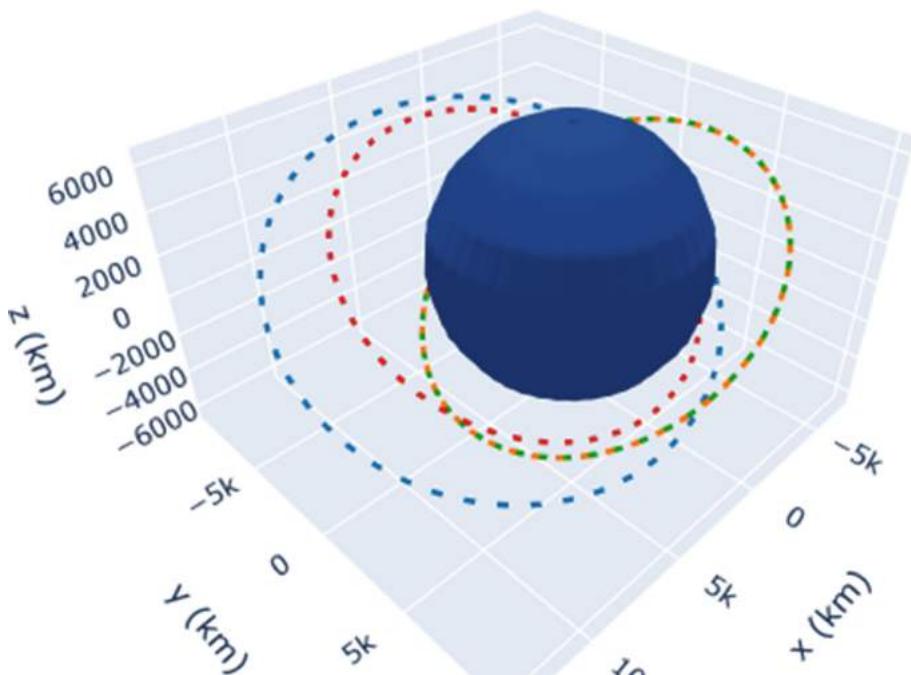


Fig. 1. Initial orbits of simulated spacecraft. Varying orbital radii and starting points were selected. The plot is three dimensional, so an angle that showed the orbits best was selected.

long string of these numbers. A 0 meant remain idle, and each number signified an impulse in the positive x, y, z or negative x, y and z directions respectively. The mating scheme for two impulse strings was devised as the following. Element for element, if the mother and father strings agree on an element, the offspring has that element as well. If they differ, a random maneuver is put in the place of that element. This scheme was selected because a scheme was needed which would allow traits of the parent solutions to be preserved in the children while still allowing for some variance for population diversity. Finally, the next generation was populated with children solutions. For evolution towards fitter solutions to occur, the fittest solutions needed to mate the most, and the least fit needed to mate the least. Evolution was carried out over 200 generations, and the fittest solutions were output as the algorithms chosen solution. To combat stagnation in fitness score, mutation and crossover were introduced. A chance of mutation was introduced such that when two sequences mated, even if they agreed on an element, there was a small chance of the element being randomly generated. To introduce crossover, a second population evolving alongside the first was created. Every five generations, these two populations would interbreed, and their traits would crossover. When the fitness scores of both populations would stagnate, a randomly generated but seeded “immigrant” population would be brought in to

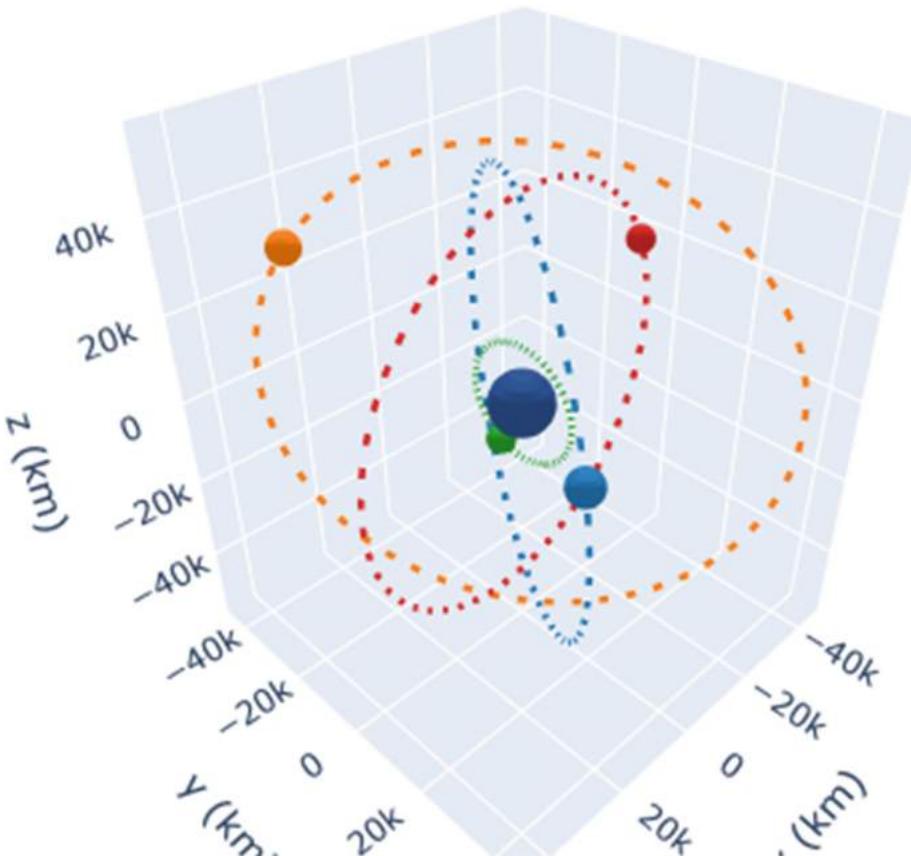


Fig. 2. Final orbits of simulated spacecraft. The only orbital parameters being selected for were inclination and eccentricity, so the orbits are polar and circular but not on the same plane or the same size. The plot is three dimensional, so an angle that showed the orbits best was selected.

mate with one of the populations. Seeded random generation means that populations were generated around an input sequence, and would have a chance of preserving the elements of the input sequence. So an “immigrant” population would be similar to the input sequence, in this case the fittest of one of the populations.

2.4 Results

The four spacecraft that were started in different orbits all achieved final orbits with fitness scores of less than 0.1, meaning the distance in radians from a polar inclination summed with the eccentricity was less than 0.1. The results can be seen in the Fig. 2, so in conclusion the genetic algorithm developed was successful

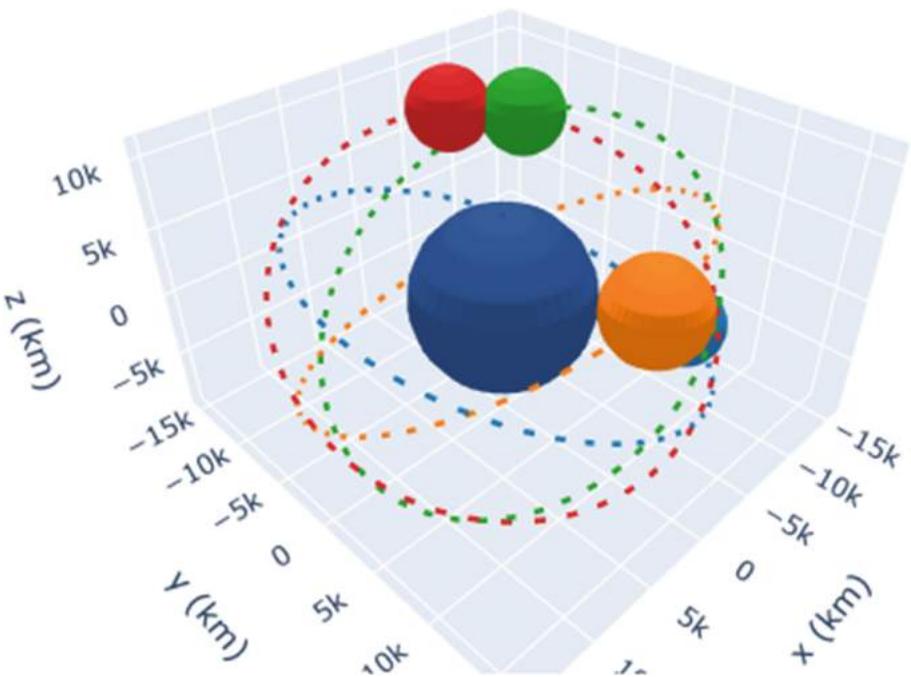


Fig. 3. Final orbits of simulated spacecraft. The orbital parameters being selected for were semi-major axis and eccentricity, so the orbits are of the same size and all circular, but not co-Planar.

at planning a series of thirty impulse maneuvers that would place the spacecraft into polar, circular orbits.

Modifications to the fitness function, and nothing else, enabled the simulated spacecraft to achieve orbits close to selected values for other pairs of orbital parameters. The final orbits when the fitness was how close the eccentricity was to zero and the semi major axis was to 15000 km are shown in Fig. 3, and the final orbits when the fitness was how close the eccentricity was to zero and the longitude of ascending node was to zero are shown in Fig. 4. The rates of convergence of the solutions were largely the same for all the orbital parameters selected, the same number of total generations were used for each and the final solutions had similar fitness scores. Initial convergence is faster than the rate of convergence late in the process, and this is to be expected. Initial populations have high population diversity, and simple changes can drastically affect fitness in a positive way. Later in the process, simple changes can still drastically affect fitness, but usually not for the better.

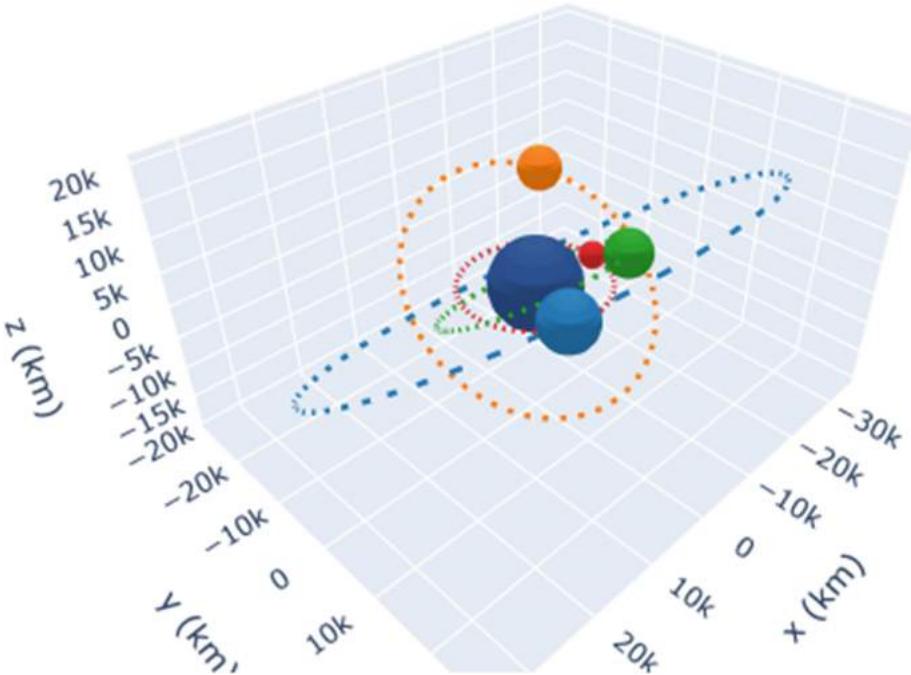


Fig. 4. Final orbits of simulated spacecraft. The orbital parameters being selected for were longitude of ascending node and eccentricity, so the orbits are circular and the right ascension of the point where the orbital plane crosses the equator is zero radians, but they are not the same size.

3 Conclusions and Future Work

In this work, we outline a constrained experimental goal of selecting a series of thirty impulse maneuvers which will result in a satellite achieving an orbit with two desired orbital parameters. We also present the genetic algorithm developed to accomplish this goal, as well as the successful results it achieves. This work was a constrained version of the more general problem of autonomous maneuver planning for spacecraft. The success of this optimization approach demonstrates a first step in the broader direction of leveraging learning systems in the coordination and autonomous direction of individual and distributed satellite systems. Future work in this line of inquiry includes achieving environmental interaction, more complex maneuvering, cooperation and coordination, multiple sequential goals, and time dependent goals of agents in an increasingly sophisticated and realistic simulated space environment using learned methods. We intend on exploring these areas in upcoming work.

References

1. Corns, S.M., Keller, J.M., Liu, D., Fogel, D.B.: Fundamentals of computational intelligence: neural networks, fuzzy systems, and evolutionary computation. *Genet. Program. Evolv. Mach.* **18**(1), 149–183 (2017)
2. Araguz, C., Bou-Balust, E., Alarcón, E.: Applying autonomy to distributed satellite systems: trends, challenges, and future prospects. *Syst. Eng.* **21**(5), 401–416 (2018)
3. Bernard, D., et al.: Spacecraft autonomy flight experience: The DS1 remote agent experiment (1999)
4. Bhaskaran, S., et al.: Orbit determination performance evaluation of the deep space 1 autonomous navigation system (1998)
5. Doyle, R.J.: Spacecraft autonomy and the missions of exploration. *IEEE Intell. Syst. Appl.* **13**(5), 36–44 (1998)
6. Hanson, J., et al.: Nodes: a flight demonstration of networked spacecraft command and control (2016)



Communicating Digital Evolutionary Machines

Istvan Elek^(✉), Zoltan Blazsik, Tamas Heger, Daniel Lenger,
and Daniel Sindely

Hungary Faculty of Informatics, 3in Research Group,
ELTE Eötvös Loránd University, Budapest, Martonvásár, Hungary
elek@map.elte.hu
<http://mapw.elte.hu/elek>

Abstract. In this article, we summarize our research results on the topic of spontaneous emergence of intelligence. Many agents are sent to an artificial world, which is arbitrarily parametrizable. The agents initially know nothing about the world. Their only ability is the remembrance, that is, the use of experience which comes from the events that happened to them in the course of their operation. With this individual knowledge base they attempt to survive in this world, and getting better and better knowledge for further challenges: a completely random wandering in the world; wandering in possession of growing personal knowledge; wandering, where evolutionary entities exchange experiences when they accidentally meet in a particular part of the world. The results are not surprising, but very convincing: without learning, the chances of survival are the worst in a world with a given parametrization. When experiences are organized into a knowledge base through individual learning, the chances of survival are obviously better than in the case of a completely random walk. And finally when creatures have the opportunity to exchange experiences when they meet in a certain field, they have the greatest chance of surviving.

Keywords: Artificial intelligence · Simulation of emergence of intelligence · Learning agents

1 Introduction

Intelligence is studied in a wide range of disciplines, such as psychology, cognitive science, philosophy, mathematics, artificial intelligence, neural networks, but is linked in many threads to evolutionary biology and even paleontology. The issue is more relevant today than ever. In the plenary presentations of neural networks and artificial intelligence conferences (Grossberg [1, 2], Adami [3], Ofria [4]), the spontaneous development of intelligence is one of the most important among the unanswered questions, more precisely, the unresolved nature of this question. How did the high level of intelligence that could be found not only in humans but also in mammals, birds, and even molluscs come into being (Gause [5, 6],

Maynard [7], Ostrowski [8]). In this context, intelligence does not mean rational thinking, but the ability that these living beings can mobilize for survival and prosperity.

Our research results are still far from these high levels of intelligence, but let us not forget that the development that resulted from the long development that led to the development of human thinking abilities, initially started from very simple thinking and action by very simple entities. We set out to simulate this simple thinking. We described the theoretical background of the work in many articles and books (Elek [9–15]).

The spontaneous emergence of intelligence is a fundamental issue for evolution. Evolutionary biologists and paleontologists already know a lot about the operation of early evolution. The first classical experiment was in 1952, where Stanley Miller was able to produce amino acids from inorganic materials with an experimental apparatus. Since then, many such experiments have taken place, and even a number of theories have been outlined to model physical and mental evolution.

2 An Artificial World

We created an artificial world that is a variation of the classic wumpus world. This is a chessboard world with dangerous energy eaters (wumpus, trap) and energy sources (gold) (Fig. 1). However, this world is different from the classical wumpus world in that it is not a real chessboard, but a torus (Fig. 2), which thus has no edges. Agents never run into boundaries, even though the world is finite. This world is the scene of the operation of artificial beings (digital evolutionary machines, workers for short). We sent them here in large quantities and watched them prosper.

3 Digital Evolutionary Machines

We have called our artificial beings (agents, workers) digital evolutionary machines that remember everything that happened to them during their lives. Their operation requires energy to be gained from the gold bars of the artificial world, meaning they need to find as much gold as possible to survive. To do this, however, they must be in motion.

There are also energy eaters in the world. One is the trap, which when workers encountered them, significantly reduces their energy. The other is the wumpus, which deprives a large amount of energy from them. If they encounter it early in life, it will surely result in their destruction. Also, the movement that is needed to find the gold requires energy of course.

The movements of the creatures are basically random. They can go from a given field in four directions (since there are no boundaries on the torus, so this is always true). They use their knowledge that whether the next field is a wumpus or not in their knowledge base. If it is, they ask for another random direction and do so until they get a harmless field. Of course, if they don't have information



Fig. 1. The world of chessboard and its players are red field (Trap), gray field (wumpus), gold field (Gold Bar). The traps and the wumpuses are energy eaters, when workers encountered them, thus the energy of the beings is greatly reduced. The gold bar, on the other hand, is a source of energy, the achievement of which increases the energy content of the creatures for helping them to survive. The numbers in the fields indicate how many steps the specified agent arrived at that field

about the next field, they can also step into an energy-eating field. If they do not perish in this, they will store this dangerous place in the knowledge base.

Those who worked so effectively that their energy level exceeds a preset value can create two (or more) offspring that are born with the same initial energy as the parent has, but inherit the knowledge that parent has accumulated so far. The knowledge of beings who have perished is also lost, unless before demise their offspring have been created who inherited the knowledge gathered so far. Workers work in three modes:

1. They do not gather knowledge, the outcome of their actions is shaped by blind chance. In the case of friendly worlds, this can also result in prosperous functioning, the creation of offspring, and population growth.
2. They gather knowledge, that is, if they have survived an encounter with a wumpus or trap, they store its place in their knowledge base. As a result, they are increasingly likely to avoid energy-eating fields. In case of large worlds,

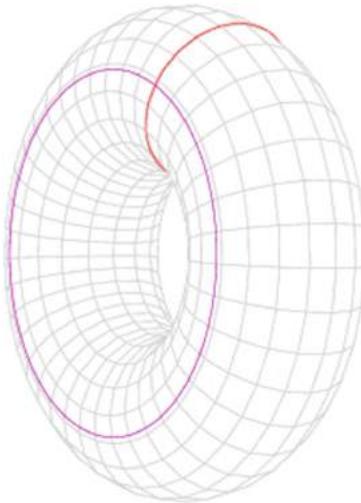


Fig. 2. The torus world is created by folding the two edges of the chessboard world (purple line on the torus) to form a cylinder. The cylinder is then folded to form a torus so that the two base circles meet (red line)

this is a protection only if they have stepped again into a known territory during their accidental wanderings.

3. They gather not only their own experiences, but they also pass them on to each other if they step into the same field at the same time. This is the coincidence.

During the work, we stored all the elements of the operation of the creatures in a database so that we could evaluate the results of the different runs.

4 Results

The success of the different simulations was measured by the change in the initial population. The increase in the size of the population and the amount of energy indicated the success of each run. The result depended on a lot of things. On the one hand, the friendliness or hostility of the world influenced the result, and on the other hand, the working mechanism of the workers. In a friendly world, non-learning populations could also be effective, especially if their replication capacity was set to greater than 2. In an unfriendly world, however, the population could no longer be effective without knowledge. The chances of populations evolving significantly increased when not only did they gather knowledge, but it also exchanged it with each other when they met in one of the fields. In a hostile world there are no chance to survive for anyone.

The following are some figures to illustrate the effectiveness of different runs. First, we present the result in not-so-friendly world. Figure 3 shows a case where

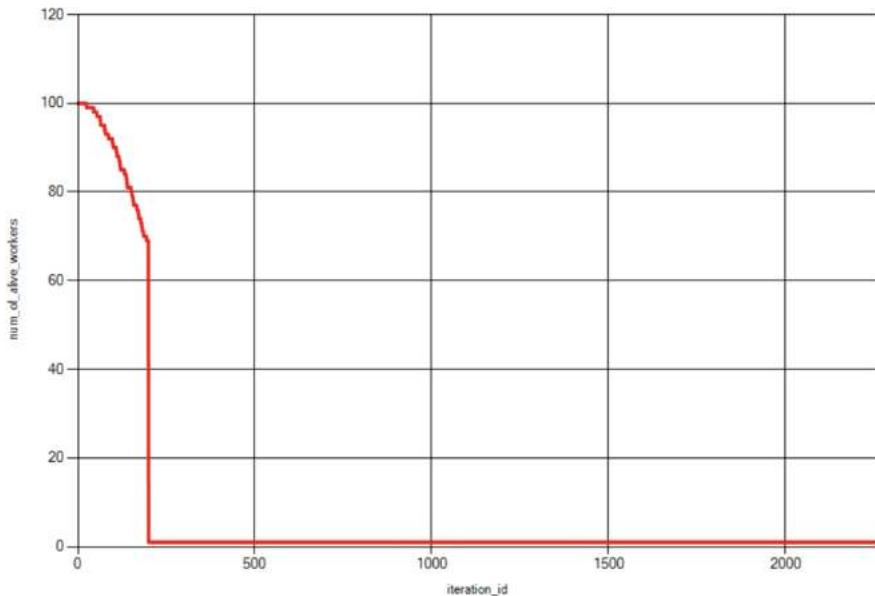


Fig. 3. A case without learning: wandering is completely random, no knowledge, no experience. Their chances of survival are not good in this hostile world where there are a lot of wumpuses and traps, but not enough gold bars. Since the number of gold-containing fields is very small, the wandering consumes their energies

we sent 100 workers into the world, but these did not gather knowledge. Due to accidental wandering, their energies ran out and they perished in a short way. For the sake of comparability, the simulation was stopped in each case in the 3000th iteration step.

Figure 4 shows a case where we also sent 100 workers into the world, but these workers had already stored the dangerous places, if they had survived such encounters. In the beginning, the devastation was great here too, because they did not yet have the knowledge to avoid dangerous fields. However, this knowledge was individual, supporting only their owner.

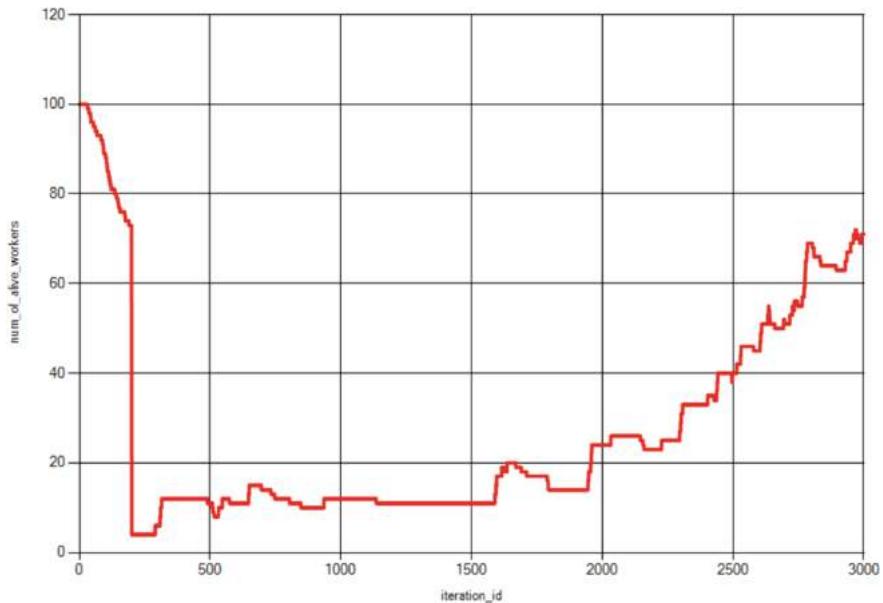


Fig. 4. Learning case: here they already gather experience, which is used in every subsequent step to decide whether a randomly selected field is dangerous or not (defensive strategy). In this way, they are less likely to step into an energy-eating field because what they already know is no longer stepped on. They thrive in a less friendly world. However, in a very energy-poor world, knowledge does not mean survival either

Figure 5 shows a case where we also sent 100 workers into the world, but these workers no longer only stored the dangerous places they encountered and survived the encounter, but also exchanged it with workers who entered the same field at the same time. Clearly, increasing each other's knowledge has increased the population's chances of survival as more and more people knew more and more about their environment.

Now let's look at a case where we are bringing in 100 workers into a friendlier world. The result is shown in Fig. 6.

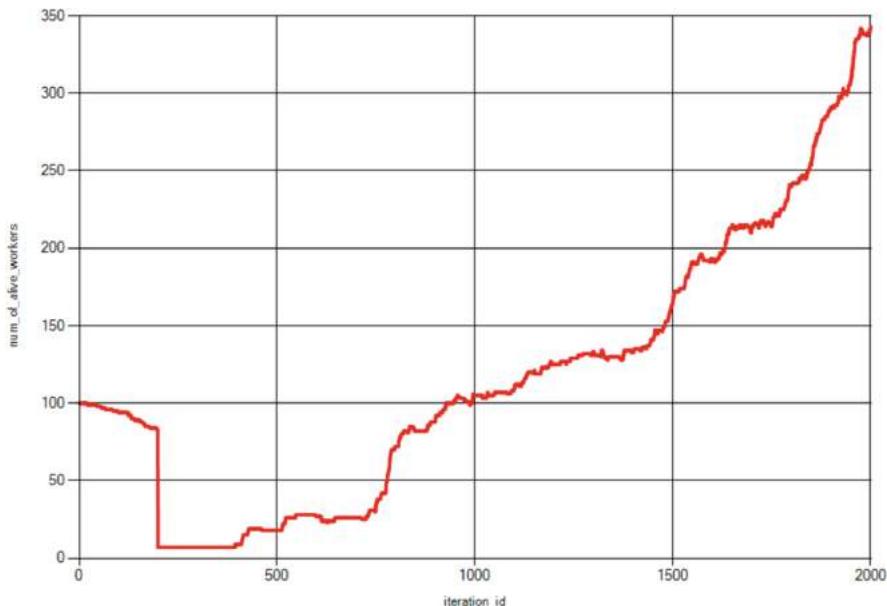


Fig. 5. A case of learner and information exchange: in this case, they gather their own experiences of dangerous fields, and when workers meet, they exchange their knowledge. This gives them a better chance of learning about dangerous fields, as a result of which they have a good chance of avoiding known energy-eating fields

In case of learning entities the friendly world also provides better conditions for population development. The result of this can be seen in Fig. 7.

If the learning workers exchange their knowledge when they coincide, their population is remarkably high in a friendlier world. The result can be seen in Fig. 8.

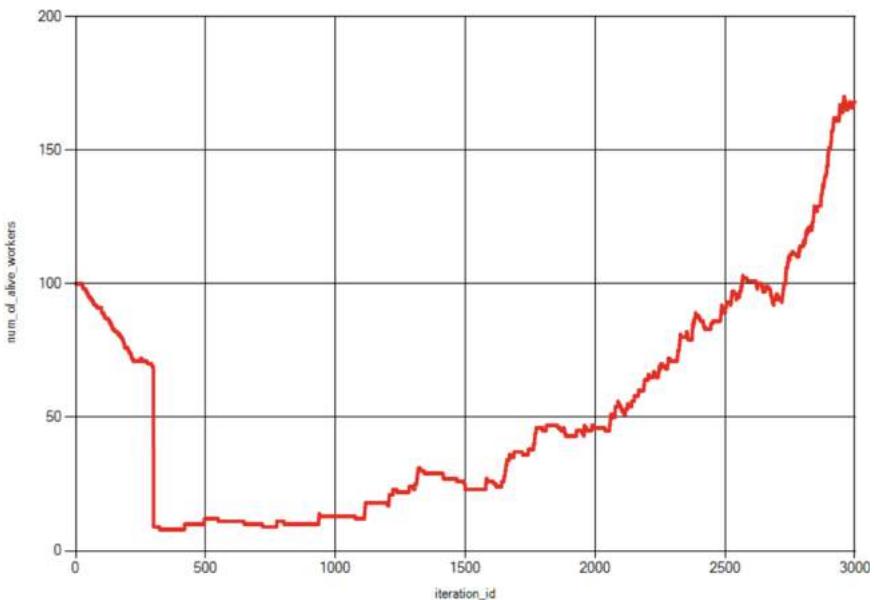


Fig. 6. The evolution of the incident-free case population in a friendly world: wandering is completely random, no knowledge, no experience gained, but the population is still surviving. Accidental wandering and the amount of energy in the world still provided them with enough energy

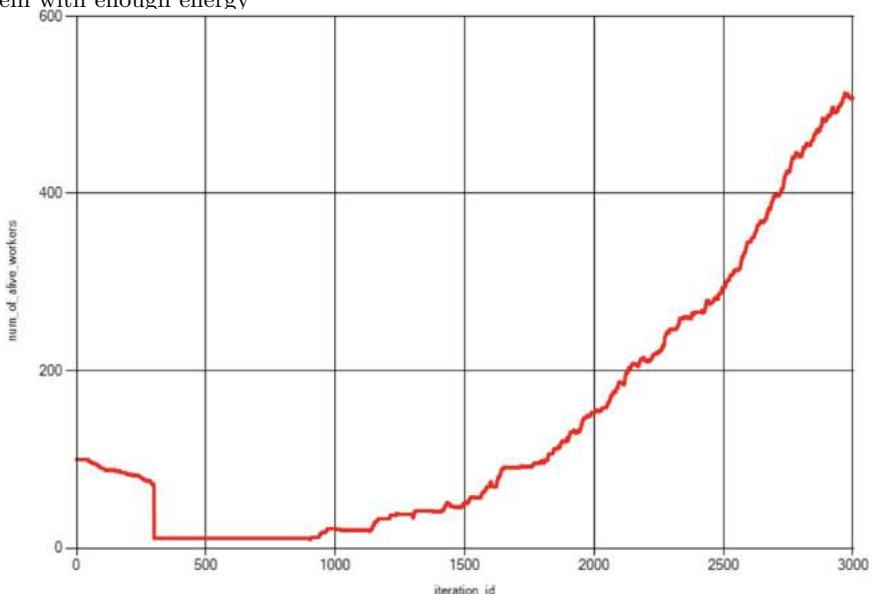


Fig. 7. Evolution of learning entities of the population in a friendlier world: they gather experience that is used in each subsequent step to decide whether a randomly selected field is dangerous or not (defensive strategy)

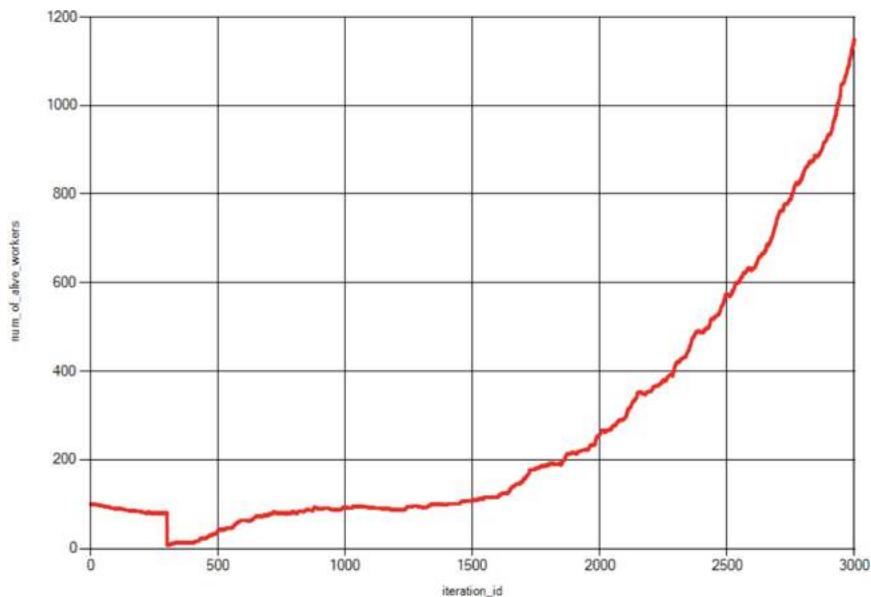


Fig. 8. The population development in case of learning and information exchanging case in a friendly world: workers gather their own experiences of dangerous fields and exchange it with each other when they meet in one of the fields. This strategy is remarkably more effective in a friendlier world

Another curiosity is how the knowledge of individuals and the population as a whole relates to each other. Compare the path traversed by one worker with the path traversed by all individuals (Fig. 9). It is clear that the knowledge of the individual remains significantly lower than of the whole population, i.e., the more times the individuals meet and exchange their knowledge, the closer they come to the knowledge of the whole population.

5 Summary

As can be seen from the above figures, the effectiveness of the population, i.e. the development of the number of people, is strongly influenced by knowledge management. Clearly, teaching each other increases your chances of survival.

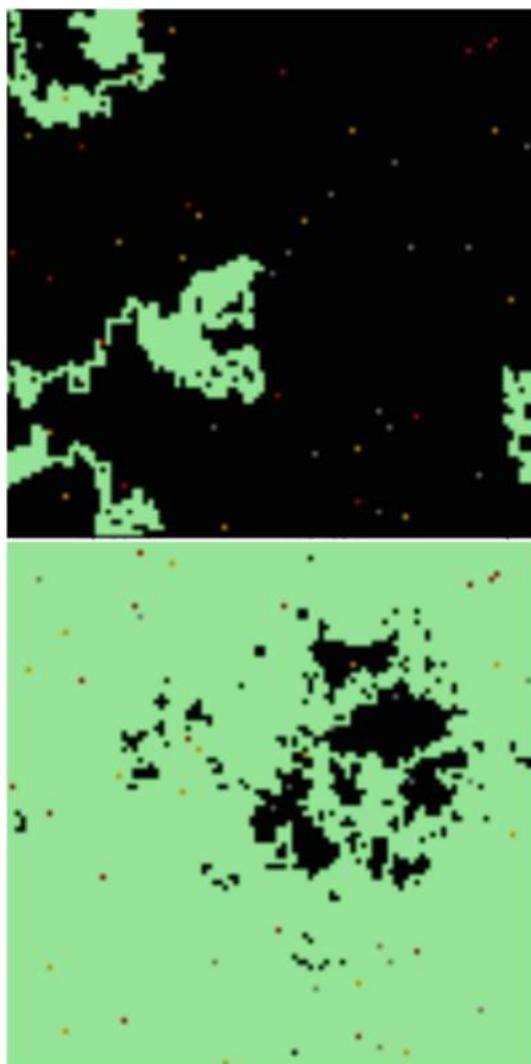


Fig. 9. Knowledge of the individual with ID '16' worker (figure on the left side) and knowledge of the entire population (figure on the right side). Dark fields are unknown, green fields are known

However, there is another important finding that is not shown in the figures. Examining the data of the workers, it can be seen that the more effective way of fighting is influenced not only by luck and the friendliness of the world, but also by the fact that the workers exist in multitude. Examining their unique data, it can be seen that during a longer simulation, almost no initial entity survives in a large world, but their experience, when incorporated into the knowledge of their descendants, contributes significantly to the “intelligence” of later generations.

Acknowledgment. The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References

1. Gail, A.: Carpenter and Stephen Grossberg: Pattern Recognition by Self-Organizing Neural Networks, MIT (Bradford Books) (1991). ISBN 0-262-03176-0
2. Grossberg, S.: Studies of mind and brain: neural principles of learning, perception, development, cognition and motor control. *Boston Stud. Philos. Sci.* **70**, 622 (2013)
3. Adami, C.: Ab Initio of Ecosystems with Artificial Life, [arXiv:physics/0209081](https://arxiv.org/abs/physics/0209081) v1 22 September 2002
4. Adami, C., Ofria, C., Collier, T.C.: Evolution of Biological Complexity, [arXiv:physics/0005074v1](https://arxiv.org/abs/physics/0005074v1) 26 May 2000
5. Gause, G.F.: Experimental studies on the struggle for existence. *J. Exp. Biol.* **9**, 389–402 (1932)
6. Gause, G.F.: The Struggle for Existence. Williams and Wilkins, Baltimore (1934)
7. Smith, J.M.: The Problem of Biology. Oxford University Press, Oxford (1986)
8. Ostrowski, E.A., Ofria, C., Lenski, R.E.: Ecological specialization and adaptive decay in digital organism's. *Am. Nat.* **169**(1), E1–E20 (2007)
9. Elek, I.: Digital evolution machines in an artificial world: a computerized model. In: International Conference on Artificial Intelligence and Pattern Recognition (AIPR-10). Orlando, USA, 12–14 July 2010, p. 169 (2010)
10. Elek, I., Roden, J., Nguyen, T.B.: Spontaneous emergence of the intelligence in an artificial world. In: Huang, T., Zeng, Z., Li, C., Leung, C.S. (eds.) ICONIP 2012. LNCS, vol. 7667, pp. 703–712. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34500-5_83
11. Elek, I.: A computerized approach of the knowledge representation of digital evolution machines in an artificial world. In: Tan, Y., Shi, Y., Tan, K.C. (eds.) Advances in Swarm Intelligence. ICSI 2010. Lecture Notes in Computer Science, vol. 6145, pp. 533–540. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13495-1_65
12. Elek, I.: Evolutional aspects of the construction of adaptive knowledge base. In: Yu, W., He, H., Zhang, N. (eds.) Advances in Neural Networks — ISNN 2009. ISNN 2009. Lecture Notes in Computer Science, vol. 5551, pp. 1053–1061. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01507-6_118
13. Elek, I.: Evolutional aspects of the construction of adaptive knowledge base. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009. LNCS, vol. 5551, pp. 1053–1061. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01507-6_118
14. Elek, I.: Principles of digital evolution machines. In: International Conference of Artificial Intelligence and Pattern Recognition, Orlando, Florida, July, 2008. ISBN 978-1-60651-000-1
15. Elek, I.: Emergence of Intelligence. Nowa Science Publishers, New York (2018)



Analysis of the MFC Singularities of Speech Signals Using Big Data Methods

Ruslan V. Skuratovskii¹ and Volodymyr Osadchy²

¹ Interregional Academy of Personnel Management, Kyiv 03039, Ukraine

r.skuratovskii@kpi.ua

² Department of Computer Science, UCF, Orlando, FL, USA

vo@it-gravity-vo.com

Abstract. The role of human speech is intensified by the emotion it conveys. The parameterization of the vector obtained from the sentence divided into the containing emotional-informational part and the informational part is effectively applied. There are several characteristics and singularities of speech that differentiate it among utterances, i.e. various prosodic features like pitch, timbre, loudness and vocal tone which categorize speech into several emotions. They were supplemented by us with a new classification feature of speech, which consists in dividing a sentence into an emotionally charged part of the sentence and a part that carries only informational load. Therefore, the sample speech is changed when it is subjected to various emotional environments. As the identification of the speaker's emotional states can be done based on the Mel scale, MFCC is one such variant to study the emotional aspects of a speaker's utterances. In this work, we implement a model to identify several emotional states from MFCC for two datasets, classify emotions for them on the basis of MFCC features and give the comparison of both. Overall, this work implements the classification model based on dataset minimization that is done by taking the mean of features for the improvement of the classification accuracy rate in different machine learning algorithms.

Keywords: Big data classification · Speech recognition · Emotion recognition · MFCC · Supervised learning · Decision trees · Mel scale features · KNN

1 Introduction

To obtain emotion score differences we use Miller function and scale. The vocal acoustics are full of emotional cues to analyze the speaker's emotional state. Since the expression of emotions occurs most often either at the beginning or at the end of a sentence, a sentence was divided by us into two parts. To the first part we refer a beginning and an end of a sentence, referred to the emotional content expressing by an author; to the second part we refer a middle of a sentence containing only the informational and narrative part. It serves to vector parameterization in KNN for speech emotions. Each emotion is associated with tone of the speaker. Emotions can be recognized both from text and sound. Each of them

Models for identifying Multiple emotional states from the MFCC.

has different approaches to identify the emotional state of the speaker. Emotions also greatly define the interpersonal relations by affecting intelligent and rational decision-making. The emotions are a communication bridge among the speaker and the listener. An interaction between individuals is way more clear and effective when the emotions are used in utterances. They play a pivotal role in engaging of a human being in a group discussion and can tell a lot about the one's mental state [1]. The information hidden in emotions ignited the process of the evolution of the speech recognition field commonly referred as automatic speech recognition. Several models of retrieval and interpretation of emotions from images of speaker's face and the recordings of his expressions, voice and tone during a conversation has been proposed by researchers. The utilization of physiological signals in the same manner has also been discussed [2]. The significance of emotions in communication can hardly be overestimated since they express the speaker's intentions to his listeners. There are several spoken language interfaces available today that support automatic speech recognition. By collecting the samples, such systems are providing a base for the speech recognition field [3]. The currently available speech systems are able of processing naturally spoken utterances with high accuracy, but the lack of emotional component makes the ASR systems less realistic and meaningful. There are several real-world fields that may benefit from the recognition of the emotional context of an utterance, such as entertainment, emotion-based audio file indexation, HCI-based systems, etc. [4].

Some of the selected features can be trained to classify, recognize and predict emotions. There are several emotions that can be extracted from the utterances. Few of the universally enlisted among them are Happiness, Fear, Sadness, Anger, Neutral, and Surprise. These emotions can be recognized by any intelligent system, constrained by computational resources. The implementation of the emotional sector of speech makes the human-computer interactions more real and efficient. The analysis of voice and speech for the sake of enhancing the quality of human conversations is reasonable and within the bounds of possibility. The results of emotion detection can be broadly applied in e-learning platforms, car-board systems, medical field, etc.

The remaining sections have the following content: Sect. 2 contains literature observation on the topic, Sect. 3 is dedicated to the description of the problem, Sect. 4 carries the details of method implementation and results achieved under problem solution and finally Sect. 5 is the conclusion. In this paper we prove efficiency of the concept of using only 12 MFCC from 39, have identified which 12 MFCC to use for speech and emotion analysis. The dataset used for this experimentation is EMODB. Variants of supervised learning approaches have been implemented to classify emotions from two databases EMODB and SAVEE.

2 Literature Review

2.1 Review of Classifications Methods

As well known KNN makes prognostications on time by quick calculating the similarity between an input sample and each training exemplar. Spectral analysis is a promising technique for detecting emotions from sample speech. Its purpose is to use a database in which the data points are separated into several classes to predict the classification

of a new sample point. Spectral analysis is a promising method of emotion detection from samples of speeches. Prosodic features of speech signals can also be used for analyzing emotions since they contain emotional information. Researchers explored the role and context of emotions by using a set of 88 features called eGmaps [5]. Speech patterns can be obtained from combination of various speech features acquired from speaker's utterances. Feature selection plays a pivotal role in the differentiation of different emotions of the same speaker from his speech [6] and it relies on selecting the best features from the signal. Different human languages have different accents, structures of sentences, and speaking styles [7] thus making the identification of emotions from utterances challenging. Various aspects of spoken languages alter the extracted features of the sound signal. It is possible that a sample speech may have more than one emotion which means that each emotion corresponds to a different part of the same speech signal, which complicates the setting of boundaries between emotions. An attempt has been made to study models of the multilingual emotion classification in literature [8].

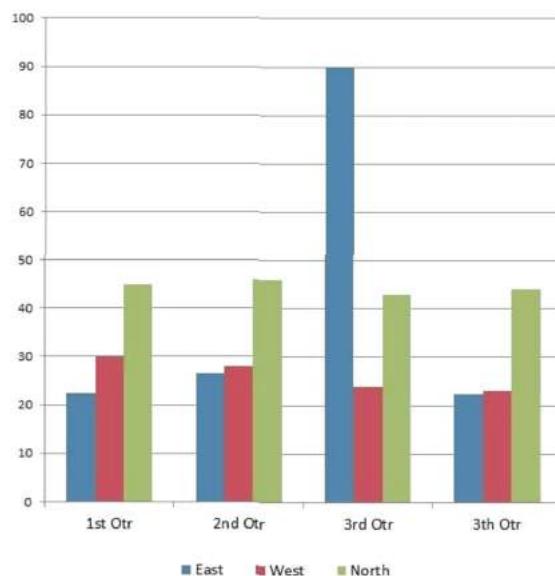


Fig. 1. Models of the multilingual emotion classification

The substantial advancement in technology has boosted the development of the emotion recognition fields [9, 10]. Call-centers and remote education for example [11]. Existing speech recognition systems can be improved by implementing spectral analysis [12].

Authors have identified classes of features extracted from speech electrography and signals. There is an important aspect of SER that includes characterization of emotional content of a speech [13]. Several speech features are obtained from speech acoustic analysis and can be used to detect and predict emotions [14]. The aim of selecting speech features is to determine properties that can improve the rate of classification emotions

from a set [15]. The machine learning models are flexible enough to adapt themselves to any model that studies emotions and show good performance in predicting tasks based on selected features [16].

Authors identified classes of features extracted from emotional speech features electrography and speech signals. There is an important aspect of SER that includes characterization of emotional content of speech [13]. Several speech features are obtained from speech acoustic analysis and can be used to detect and predict emotions [14]. The aim of selecting speech features is determination of properties that can improve rate of classification for emotions from a feature set [15]. The machine learning approaches are flexible enough to adapt themselves to any model that studies emotion and show good performance perform well predicting tasks based on selected features [16].

2.2 Maintaining the Integrity of the Specifications

Emotions are intertwined with mood, temperament, personality, sentiment and motivation [17]. Emotions can be understood as a complex feeling of the mind that results in physiological and physiological changes. Human thoughts and behavior is influenced by emotions and there are instances of changes in body when it encounters different emotional states [18]. Literature [19, 20] proves that there is a considerable influence of emotional syndromes on human actions and reactions. Several applications created by researchers rely on emotion detection as an integral component for identification of behavioral patterns [21, 22]. Speech features can be extracted from various sources to accomplish predictive analytics (Fig. 1). The list of sources includes vocal tract, excitation source and prosodic extraction.

Table 1. Speech preprocessing techniques

SER [23]-Speech Emotion Recognition System (Component and tasks)	
Feature Extractor	Emotion Classifier
Takes signal input and generates emotion feature	Map the speech with one or more emotions

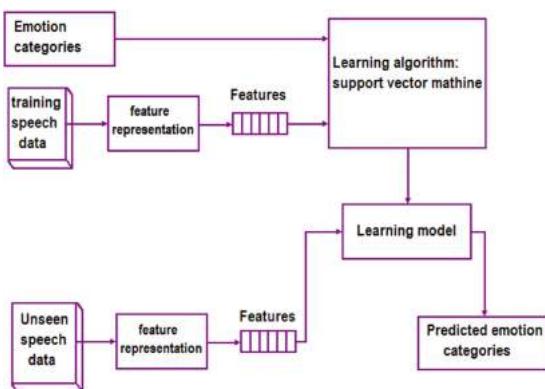


Fig. 2. Framework for supervised emotion classification [23].

Emotions can be broadly studied using both discrete and continuous approaches. Different classes represent different kinds of emotions and the continuous approach of studying of emotions is a derivation based on combination of several psychological measurements on different axes [24]. Speech Emotion Recognition in the main identifies emotions on the basis of categorical approach as it shown in Table 1, which depends on usage of common words. Researchers derive emotions from expressions of face, speech and various physiological signals. The analysis of facial expressions is a great way to find emotions [25–29] since human face displays emotions very aptly even without a single word uttered. Voice recordings are potentially important for expressing the speakers' mental state and their intentions. Speech features (Fig. 2) can be studied as vectors for detection of emotion from a data set. [30–32]. The autonomous nervous system allows to assess an emotion and thus utilize physiological signals like ECG, RSP BP to recognize people's emotions and possibly help cure mental illnesses [17].

We took into consideration the line of temperament. We also integrate the state in which a representative of this type of temperament may be. There are 8 axes of Miller are applied by us. Also it is optimal to take 8 axes.

Consciously controlling the volume level, for example, to emphasise a secret, even an angry person can make the voice calmer and quieter in order to show that it was a secret, a question or the essence of the issue. On the contrary, in order to highlight the characterisation of some hero of his story, the speaker can consciously increase the amplitude of the voice.

Physiological features of voice and hearing. It takes into account the average amplitude (frequency) and other characteristics of the person's voice, obtained on the basis of data from the database.

Given the above assumptions, if we wish to approach the study of temperament, we first need an understanding of emotion. Such understanding is not easy to come by a glance at a typical textbook of psychology will show that "emotion" is used to refer to a ragbag of apparently disconnected facts and is never itself clearly defined at all. Yet, within one branch of psychology, namely, animal learning theory, there has long been a reasonably clear consensus that emotions consist of states elicited by stimuli or events which have the capacity to serve as reinforcers for instrumental behavior. This, for example, is the framework within which Miller analyzed the concept of "fear" and its role in avoidance learning [73]. The term temperament has considerable overlap with dimensions of personality and with emotional, cognitive, and behavioral functions.

3 Statistical Data Corpus

3.1 Emotional Speech Databases

There is need of suitable databases to train the emotion recognition systems. Researchers suggest several existing databases aligned with the task of detecting and classifying the emotions. These data bases can be categorized into three broad domains that cover acted emotions, natural emotions and felicitated emotions [33]. Out of the three mentioned domains enacted emotions are frequently supported by research since they are strong and reliable. EMODB is one of such highly used database for emotional classification. SAVEE is yet another enacted emotion database used for studying emotions. EMODB is

a Berlin database for emotions while SAVEE is an English database specifying various emotions.

Three databases have been utilized for training and testing:

1. BERLIN DATABASE
2. SENTIMENT DATABASE English
3. AVEE DATASET

Berlin Database was created in 1999 and consists of utterances spoken by various actors. EMODB has different number of spoken utterances for seven emotions [34]. Emotions included in the database are anger, boredom, disgust, fear, happiness, indifference, sadness. The dataset contains more than 500 utterances spoken by 51 male and 60 female actors from the age 21 to 35 years. The emotions labelled in EMODB are listed in Table 2.

Surrey Audio-Visual Expressed Emotion (SAVEE) [35, 36]. The increasing demand of research in speech analysis led to the development of SAVEE database recordings to help study automatic emotion recognition system. The database contains recordings from 4 male actors in 7 different emotions, 480 British English utterances in total.

TIMIT corpus was used for sentence selection and contains phonetically-balanced emotions. The data were recorded in a visual media lab with efficient audio-visual equipment. The recordings were then processed and labelled. 10 subjects under audio, visual and audio-visual conditions evaluated quality of performance, of the recordings. The actors of the database utterances were four male speakers annotated as DC, JK, JE, KL. The speakers who contributed for recordings were postgraduate students and researchers at the University of Surrey. The age of speakers lies between 27 to 31 years. Seven discrete categories of emotions described as anger, disgust, fear, happiness, sadness and surprise were recorded [37]. The focused research was carried out on recognizing the discrete emotions [38]. Table 3 compares the features of both the datasets used in experimental analysis.

Table 2. EMODB labels

Letter	Emotion (German)	Emotion (English)
W	Ärger (Wut)	Anger
L	Langeweile	Boredom
E	Ekel	Disgust
A	Angst	Fear/Anxiety
F	Freude	Happiness
T	Trauer	Sadness
N	Neutral	

Table 3. Comparison of EMODB and SAVEE

Attributes	EMODB	SAVEE
No. of speakers	111	4
Age of Speakers	21 to 35 years	27 to 31 years
No. of utterances	500+	480
Language	German	British English
Emotions	Angry, happy, anxious, fearful, bored disgusted, neutral	Anger, disgust, fear, sadness, happiness, surprise, neutral

3.2 Feature Extraction [39–41]

The core step for recognizing speech or emotions from speech is extracting the features of speech. The process of feature extraction refers to identifying the components of the vocals from the audio signal. The audio signal is good source of linguistic information if the noise is discarded in the signal. There is another interpretation of feature extraction which says that it is characterization and recognition of information specifically related to the actor's (speaker) mood, age, gender. The general process of feature extraction involves transformation of raw signal into feature vectors, which suppress the redundancies and emphasize on the speaker specific properties. The properties are like pitch, amplitude, frequency. The speaker dependencies such as health, voice tone, speech rate and acoustical noise variations may vary the speech signal during the testing and training sessions due to [31, 42, 43]. The shape of the vocal tract filters the sounds generated by human beings which if determined efficiently can be used to derive phoneme representation of the speech sample with high accuracy.

The features to be extracted from speech can be studied under three categories named as High level Features which may include phones, lexicon, accent, pronunciation; Prosodic and Spectra-temporal Features that can be studied as pitch energy duration, rhythm, temporal features) and Short term spectral and prosodic Features pertaining to spectrum glottal pulse [44–46]. Short-term spectral features aid in better prediction with higher accuracies for various applications. The spectrogram analysis can be used for information extraction from the short term spectral features. Linear Predictive Cepstral Coefficients (LPCC), Mel-Frequency Discrete Wavelet Coefficient (MFDWC), Mel-Frequency Cepstral Coefficients (MFCC) are most commonly used short term spectral features for speech analysis [31, 42, 47].

3.2.1 MFCC

MFCC are considered as the commonly used acoustic features for the task of identifying the speaker and the properties of the speech. MFCC takes into account human perception sensitivity with respect to frequencies. The combination of both is best for speech identification and differentiation. The importance of MFCC is inspired by the fact that the shape of vocal tract that includes tongue, teeth, throat etc. filters the sound generated by human speakers. The accuracy in determining the shape enables easy analysis of the

sound that comes out through the vocal tract. The accurate in determination of the shape of sound can help in finding the phonetic information. The task of MFCC is to accurately represent envelop of the short time power spectrum of the sound when it traverse through the vocal tract [41, 47, 48].

Mel Frequency Cepstral Coefficients (MFCCs) were identified as a feature and is widely applicable to automatic speech recognition and identification of speaker. The correlation among the actual and heard signal frequencies can be derived efficiently by incorporating Mel scale. Davis and Mermelstein were pioneer in identifying MFCC as sound feature in the 1980's. MFCC ever since its discovery has been considered important feature for analysis of speech signals. There are few other features along with MFCC, like Linear Prediction Coefficients (LPCs) and Linear Prediction Cepstral Coefficients (LPCCs) that were coined before MFCC and remained the main features for automatic speech recognition (ASR), especially with classification algorithm such as HMM [49]. In practice 8 to 12 or 13 MFCC are considered for representing the shape of spectrum and hence are used for speech analysis [50]. MFCC are highly preferred choices in automatic speech recognition systems [51]. Authors found that MFCC is effective for end to end acoustic modelling using CNN [52, 53]. MFCC is widely used feature while considering speech modelling [54, 55]. MFCC based comparative study of speech recognition techniques was conducted by authors who found that MFCC with HMM gave recognition accuracy of 85 percent and with deep neural networks the score was 82.2 percent [56]. Computation of MFCCs includes a conversion of the Fourier coefficients to Mel-scale [57]. Mel-scale are the most popular variant used today, even if there is no theoretical reason that the Mel-scale is superior to the other scales [58].

3.3 Decision Tree Classifiers

There are several machine learning algorithms that can be applied for recognizing speech emotions. The algorithms can be used independently or in hybrid mode for classifying emotions. Decision tree are one of the machine learning algorithms that can be used for classification task [59, 60]. The Decision tree uses the supervised learning approach that works on labelled data. The data is split into train and test subsets for carrying out the classification task. The current work uses Random forest, KNN and XGBoost algorithms for classifying emotions. All the mentioned algorithms are the variations of decision tree classifiers and a brief description of each classifier is given below [14].

3.3.1 Random Forest

One of the most flexible and easily implementable learning algorithm in machine learning is Random Forest. The algorithm provides better solutions over basic decision trees. The random forest depends on few parameters which if tuned can provide good results. The algorithm is widely used due to its simple and flexible aspect of implementation. Random Forest supports both regression and classification task while modelling a solution. It is supervised learning technique that creates random forests. The ensemble decision trees are referred to forest and mostly use bagging for training [51, 52]. The importance of regressive bagging lies in the fact that it increases the overall results. Multiple decision trees are build and together to increase efficiency in random forest algorithm.

Random forest generation uses same hyper parameters that are used for decision tree or a bagging classifier. The class of classifier does not require combining the decision trees to bagging classification algorithm. The algorithm proceeds by searching for the best feature from available features subset. The selected feature will then be used for splitting the node. The node split diversifies and enhances the results. The relative importance of each feature is measured while prediction. SK-learn tool can be used to measure a feature importance. The tool reduces impurity at the tree nodes that use the feature, across all trees in forest. Score for each feature is automatically computed after training. Features and observations are randomly selected by random forest and averaged for building several decision trees. The decision uses rules and facts for decision making and trees from over fitting. Random forest prevents it by creating subsets and combining them to subtrees. The only limitation of random forest is slow computation which is affected by number of trees build by random forest [61].

3.3.2 XGBOOST

XGBoost [61] uses gradient boosting technique to ensemble decision trees. XGBoost is stands for “eXtreme Gradient Boosting”. Small, medium structured and tabular data uses XGBoost for classification. XGBoost is studied as improvisation upon the base GBM framework. Optimization and algorithmic techniques are used to improve the base framework of GBM. Regularization is used to enhance the performance of algorithm by preventing data overfitting. The algorithm automatically learns best missing values depending on the training loss and handle variety of patterns of sparsity more efficiently. It also has built-in cross-validation method at each iteration.

XGboost is sequential tree building algorithm implemented by parallelization. The interchangeable nature of the loops determine the base of building algorithm. The external loop is responsible for maintaining the tree count, and features are calculated by the internal loop. Loops are interchangeable and thus enhance run time performance. All the instances are globally scanned, initialized and sorting is done using parallel threads. This switch of loops increases the algorithmic performance. The parallelization overheads in computation are offset. The tree splitting within GBM framework for stopping the split is greedy in nature. Splitting of tree at node depends on the negative loss criterion at the point of split. XGBoost uses ‘max_depth’ parameter as specified instead of criterion first, and pruning of the trees is done backward. The computational performance is improved significantly by using this ‘depth-first’ approach improves [61].

3.3.3 Sklearn KNeighbors Classifier and KNN

The early description of KNN was found in 1950. KNN is labour intensive approach for large datasets. It was used for pattern recognition initially. The learning of KNN is based on the comparison test data with train data such that both have similarities. A set of N attributes describe the tuple data. An n- dimensional space is used to store all the training tuples where each of them corresponds to a point in space. The pattern space for k training tuples that are closest to unknown tuple is identified by the K-nearest neighbor classifier. The closest found points are referred to nearest neighbors and euclidian distance defines the nearness of the neighboring clones [62].

The Euclidean distance between two Co tuples represented by $A_1 := \{a_{11}, a_{12}, \dots, a_{1n}\}$, $A_2 := \{a_{21}, a_{22}, \dots, a_{2n}\}$ is obtained using following calculation,

$$d(A_1, A_2) = \sqrt{\sum_{i=1}^n (a_{1i} - a_{2i})^2} \quad (1)$$

And in case of parametrized KNN we use in particular case the following generalization of formula (1):

$$d(A_1, A_2(y)) = \sqrt{\sum_{i=1}^n w_i (a_{1i} - a_{2i})^2}$$

After we choose a class y maximizing this distance. Weights depend on the neighbor's number $w(x_i) = w(i)$. For brevity, we denote these quantities by w_i . In general case we utilize some realizations of classification by formula

$$a(x) = \arg \max_{y \in Y} [x(i) = y] w_i, \quad (2)$$

where $x(i)$ are points near testing point y and Y is assumed by us class of object y .

Thus the difference of values of attribute in A_1 and A_2 is obtained. The difference is then squared to accumulate total distance count. Attributes with large ranges can outweighs attributes within small ranges (binary attributes).

To normalize data, we will use Z-scaling based on the mean value and standard deviation; dividing the difference between the variable and the mean by the standard deviation. In practice, minimax scaler and Z-scaling have similar applicability and are often interchangeable. However, when calculating the distances between points or vectors, Z-scaling is used in most cases. And the minimax is useful for visualization, for example, to transfer the features encoding the color of the pixel into a range of [0... 255] [2].

Recall that Z-scaling based on the mean and standard deviation is dividing the difference between variable and mean by standard deviation;

$$z = \frac{x - \mu}{\sigma} \quad (3)$$

where μ is an expected value and σ is the standard deviation of the value.

Because the K-Nearest Neighbours Algorithm (hereinafter the KNN algorithm) is about the distance from a point to a class, Z-scaling is usually used for its application, as it is known that in calculating the distances between points or vectors in most cases the result of Z-scaling is much more accurate.

The main advantage of Z-scaling is that it preserves the normal distribution of a random value.

Z-normalization keeps a distribution normal if it was, and a non-normal distribution converts to a non-normal distribution too. As a result of such a transformation, we get a value with 0-th mean (mean) and 1 dispersion.

Normalization is applied to each attribute value to resolve the issue.

Most common class is assigned to unknown tuple among its k-nearest neighbours. If the value of K equates 1, the unknown tuple is assigned the class of the training tuple that is closest to it in pattern space. Real value prediction is returned by KNN for unknown value tuples. The unknown values the classifier of KNN returns the average of the real valued labels associated with K-nearest neighbours of unknown tuple [62–64].

Min-max scaler keeps outliers so we have to use robust scaler of statistics that are robust to outliers. As an alternative normalization we propose to use Z-normalization (Z-scaler). This normalization holds a normal distribution.

4 Problem Statement

Authoritative literary sources mention MFCC as an important feature for analyzing and classifying various aspects of speech. Some of them state that only 13 MFCC features are sufficient enough to be considered for experimentation. There is currently no experimental validation for this statement. Moreover, there is no sufficient research on the identification of these 12 MFCC from the extracted 20 base features of MEL scale. MFCC also have derivatives of base features named as delta and double delta. The aim of the work is to establish experimental proof of considering only 8 to 13 MFCC from extracted 39 features of MFCC [23, 50]. The current work conducts experimental analysis on MFCC obtained from EMODB, a Berlin database that consists of more than 500 utterances, which were recorded from 111 both male and female speakers from various age groups.

5 Experimental Confirmation of Results

To reach more effectiveness we utilize parametric KNN method. To present a number row, you have to look at the dynamics of the signal change. In large sentences, emotions are placed either at the end of the sentence or at the beginning. Therefore, when parameterizing the distance vector, it is important to set weighting factors in such a way as to distinguish the significance of the start of the sentence distance and the distance to the end of the vector coordinate.

When applying discriminant recognition techniques to the object, the feature vector is displayed in five-dimensional space. The training matrix will be defined as the data matrix:

$$W = \begin{pmatrix} w_{11} & w_{12} & w_{13} & w_{14} & w_{15} & w_{16} \\ w_{21} & w_{22} & w_{23} & w_{24} & w_{25} & w_{26} \\ w_{31} & w_{32} & w_{33} & w_{34} & w_{35} & w_{36} \\ w_{41} & w_{42} & w_{43} & w_{44} & w_{45} & w_{46} \\ w_{51} & w_{52} & w_{53} & w_{54} & w_{55} & w_{56} \end{pmatrix}$$

We highlight five characteristics: in addition to the three mentioned in the beginning, we will highlight the beginning of the sentence and its end, as they are emotionally loaded. They carry not only an information load, but also an emotional one. When analyzing the

beginning and the end of a sentence, we will take into account how much each feature has changed relative to the average characteristics of the speaker. Therefore, forming a data-vector with 5 coordinate we substitute in the i -th coordinate characterizing the amplitude the ratio $\frac{a_i}{M(a)} = v_i$ of the amplitude of the current phrase a_i to the average value $M(a)$ of the amplitude of the person's voice.

The columns of the matrix W contain the weighting factors [66, 67] that most characterize this class of emotion, and the rows contain the features extracted from the phrase. For example, high amplitude is most characteristic for the emotion of aggression. The corresponding weight coefficient in the aggression column will be larger. Knowing the average values of the features of speech, we construct the matrix based on the changes in the parameters.

The following simulations and experiments were performed. The flow graph in Fig. 3 shows the steps carried out for the experiment.

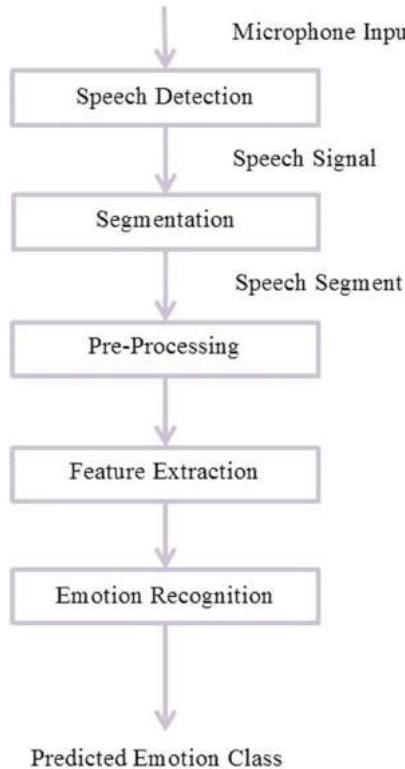


Fig. 3. Flowgraph of the emotion classification using decision tree classifiers.

The experiment was done on MFCC extracted from the EMODB and SAVEE data set (Table 4). Four subsets of MFCC features comprising 20 cepstral constants were analyzed for feature importance. This was done to identify which 13 MFCC should be used for speech emotion analysis. The result of each subset using different classifiers areas is shown in Table 5. Each subset was used to classify emotions using supervised learning algorithm (variants of decision trees). It was observed that the results obtained using 20 MFCC over set of first 13 was very near to each other. There was no effective and substantial difference in the accuracy scores for classification while using 13 and 20 MFCC subsets. Increased number of features often increases the complexity of the system and so if 13 MFCC are used instead of 19 MFCC the results will not suffer much loss. The validation of the experiment was also done by extracting important features from PCA Analysis It was seen that most of the important features corresponded to initial 13 MFCC extracted from dataset. The accuracy score of classification on EMODB using M0-M12 was 52%, 50%, 47%, 41% using subsets M0-M19, M10-M12, M15-M17 and M16- M19 respectively using Random Forest classifier. KNN shows 56%, 44%, 45% and 42% accuracy score using subsets M0-M19, M10-M12, M15-M17 and M16- M19 respectively. XGB showed poor performance on the original extracted dataset for classification task. SAVEE results as shown in Table 6 depicts accuracy scores of RF using subsets M0-M19, M10-M12, M15-M17 and M16-M19 are 57%, 55%, 50%, 50% respectively. For KNN the performance of four subsets is 67%, 54%, 55%, 50%.XGB showed poor performance on all the subsets. The results obtained in Table 5 and Table 6 clearly indicates that selecting M0-M12 would be a better choice for features from MFCC data set. It can be seen from the results that first 13 Mel coefficients can successfully be used for playing with speech over using 20 features. This selection shall only optimize the results but also reduce the complexity of the model [72] thereby reducing the computation time.

Table 4. Label encoded emotions for EMODB

Emotion EMODB	Emotion SAVEE	Encoded label
Fear/Anxiety	Anger	0
Disgust	Disgust	1
Happiness	Fear	2
Boredom	Happiness	3
Neutral	Sadness	4
Sadness	Surprise	5

There after the dataset was minimized and re-experimented for feature importance and classification. The accuracy of results for classification increased effectively but the set of important features still contained features M0 to M12 that initial 13 features. The reason behind this is that as the sound signal passes through the vocal tract and comes out as the utterance there is a subsequent addition of noise to the originally generated signal. Addition of noise disturbs the energy whose log is computed as the base of MFF extraction. The induction of noise imputes the signal at later levels so the original signal remains intact for usage in analaysis [65, 68–71]. The features present here can be successfully used for better results as compared those extracted towards the end of the sample of each speech signal .

Table 5. Results of EMODB with Subsets of MFCC

MFCC	M0-M19	M0-M12	M15-M17	M6-M19
RF	52%	50%	47%	41%
KNN	56%	44%	45%	42%
XGB	40%	38%	28%	27%

Table 6. Results of SAVEE with Subsets of MFCC

MFCC	M0-M19	M0-M12	M15-M17	M6-M19
RF	57%	55%	50%	50%
KNN	67%	64%	55%	50%
XGB	30%	38%	30%	30%

The results in Table 5 and Table 6 show that M0-M19 and M0-M13 has nearly similar results for accuracy on classifiers. The datasets in Table 5 and Table 6 used the non-manipulated MFCC extracted from the speech utterances in in EMODB and SAVEE.

For further analysis the datasets were minimized and preprocessed using min max scaling. The important features identified for the minimized data using principle component analysis are in Fig. 4 and Fig. 5. The x-axis of the plot shows various classes of emotion and y axis plot shows the MFCC features.

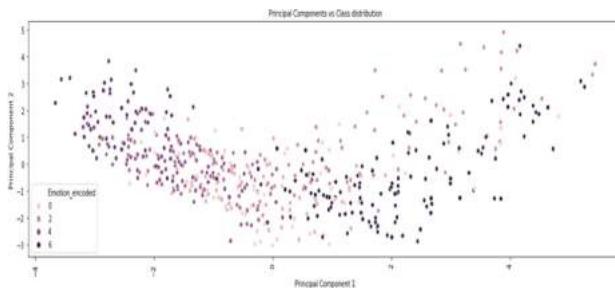


Fig. 4. Principle Component Vs Class Distribution for EMODB.

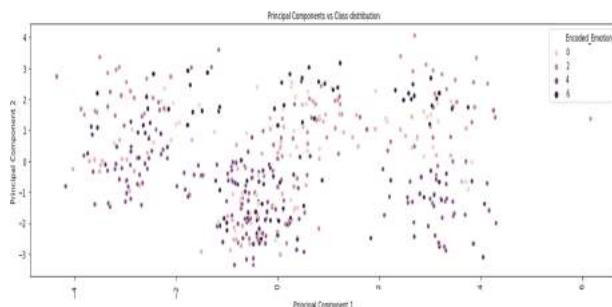


Fig. 5. Principle Component Vs Class Distribution for SAVEE.

The results of using 13 MFCC from minimized datasets EMODB and SAVEE are shown in Fig. 6(a), (b), (c). SAVEE results can be seen in Fig. 8(a), (b), (c). The plots in the mentioned figures clearly display the precision; recall and F1-scores for emotions in both the datasets using variants of decision trees.

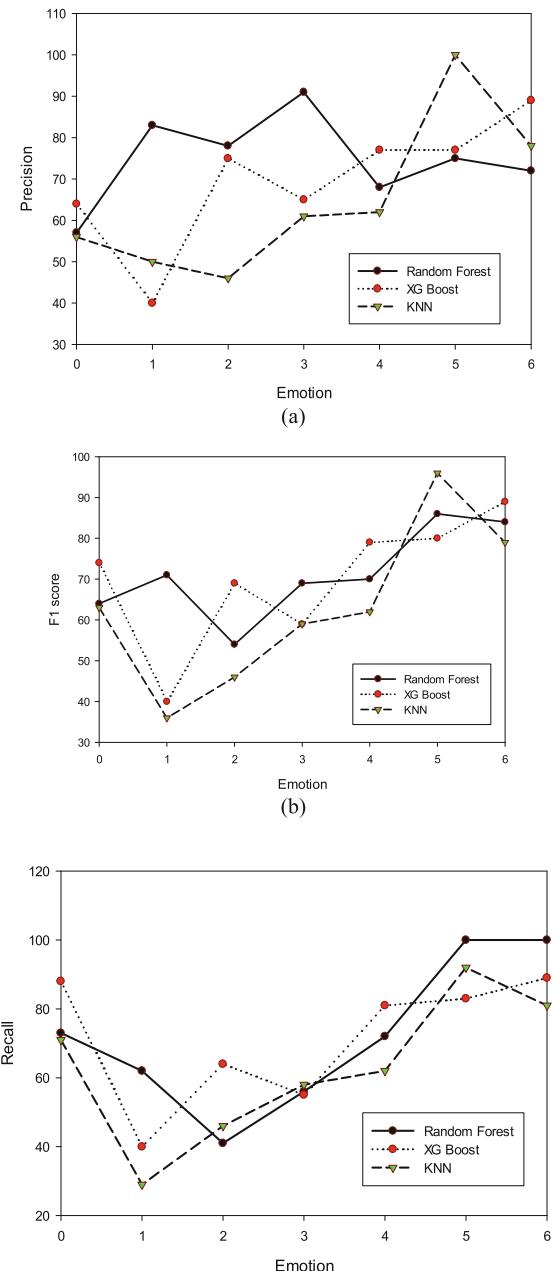


Fig. 6. (a) Emotion Vs Precision Score for First 13 MFCC using KNN, Random Forest and XGB Classifiers on EMODB (b) Emotion Vs F1-Score for First 13 MFCC using KNN, Random Forest and XGB Classifiers on EMODB. (c) Emotion Vs Recall Score for First 13 MFCC KNN, Random Forest and XGB Classifiers on EMODB.

Results showed that for EMODB all the three classifiers defined fairly variable results. Boredom (91%), anger (74%) and sadness (100%) had highest precision for random forest XGB and KNN respectively in EMODB. For SAVEE a higher precision rate for Disgust (84%) was identified using random forest. KNN and XGBoost identified sadness more precisely over remaining six emotions where the scores for sadness were 84%, 70% and 74% with KNN, random forest and XGBoost respectively (Fig. 7). A common conclusion was obtained from the results of both the datasets that sadness was commonly identified with highest precision using KNN (Fig. 9). So KNN can be effective for studying the emotion sadness in emotion analysis.

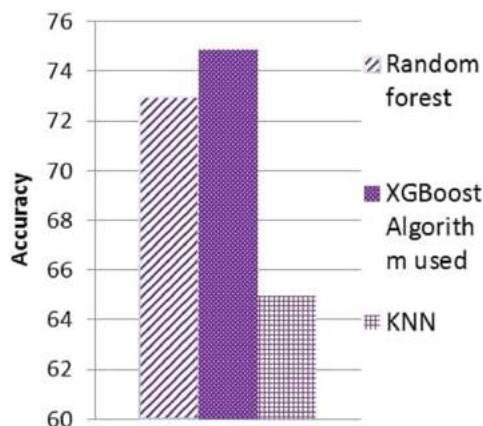


Fig. 7. Accuracy score for emotion classification using KNN, random forest and XGB classifiers on EMODB.

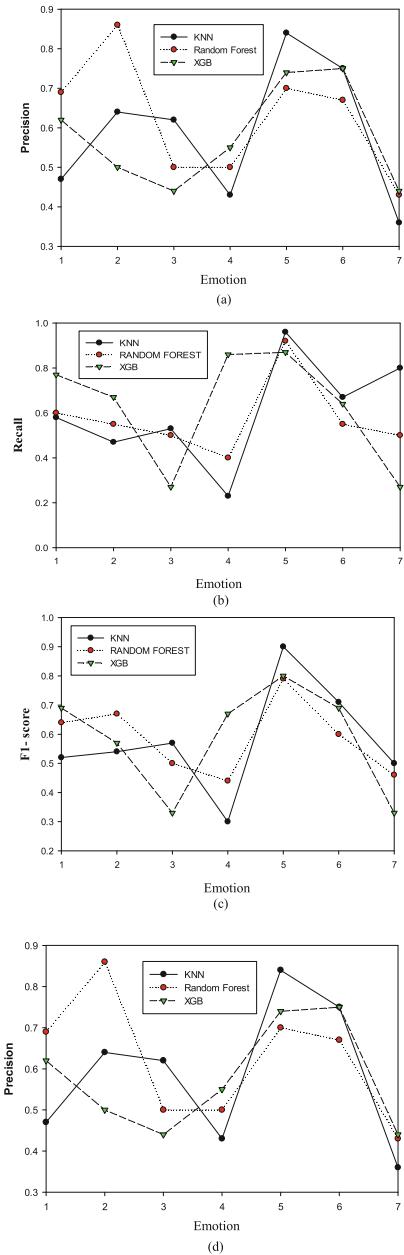


Fig. 8. (a) Emotion Vs Precision Score for First 13 MFCC using KNN, Random Forest and XGB on SAVEE. (b) Emotion Vs Recall Score for First 13 MFCC using KNN, Random Forest and XGB on SAVEE. (c) Emotion Vs F1 Score for First 13 MFCC KNN, Random Forest and XGB on SAVEE. (d) Emotion Vs Precision Score for 13 MFCC using Three Decision Tree Classifiers on EMODB.

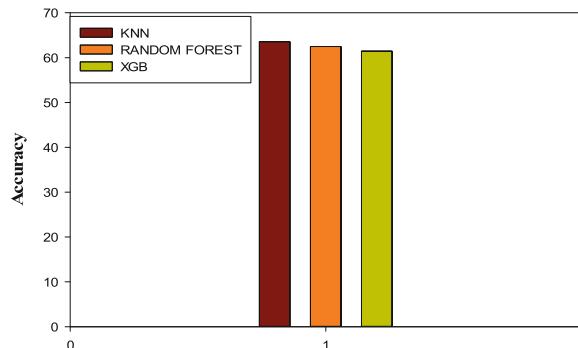


Fig. 9. Accuracy score obtained with First 13 MFCC using KNN, random forest and XGB.

6 Conclusion

Speech features have always remained the one of the regressively studied topic in research. Speech or utterances contain vital information regarding the intention, emotion and psychology of the speaker. The paper studied the use of one such speech feature called MFCC and utilized it to classify emotions using two datasets. The work also tried to establish the importance of using first 13 MFCC when we have a set of 20 Mel constants that can be extracted for speech based on the vocal physiology of human mouth. Accuracy scores for emotion classification using variants of decision tree approach have been obtained for EMODB and SAVEE for two datasets. KNN was identified as the common classification algorithm for both datasets. The score of sadness as obtained from KNN were highest for both the datasets (Fig. 7). The results of the experiments can be utilized for predicting emotions and personality of the speaker [68]. The results can be integrated with various applications pertaining to human psychology and medical treatments.

References

1. Koolagudi, S.G., Rao, K.S.: Emotion recognition from speech: a review. *Int. J. Speech Technol.* **15**(2), 99–117 (2012)
2. Marechal, C., et al.: Survey on AI-based multimodal methods for emotion detection. In: Kołodziej, J., González-Vélez, H. (eds.) *High-Performance Modelling and Simulation for Big Data Applications*. LNCS, vol. 11400, pp. 307–324. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-16272-6_11
3. Sreenivasa Rao, K., Koolagudi, S.G., Vempada, R.R.: Emotion recognition from speech using global and local prosodic features. *Int. J. Speech Technol.* **16**(2), 143–160 (2013). <https://doi.org/10.1007/s10772-012-9172-2>
4. Koolagudi, S.G., Devliyal, S., Barthwal, A., Rao, K.S.: Real Life emotion classification from speech using Gaussian mixture models. In: Parashar, M., Kaushik, D., Rana, O.F., Samtaney, R., Yang, Y., Zomaya, A. (eds.) *Contemporary Computing. IC3 2012. Communications in Computer and Information Science*, vol. 306. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32129-0_28

5. Latif, S., Rana, R., Younis, S., Qadir, J., Epps, J.: Transfer learning for improving speech emotion classification accuracy. In: Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH, vol. 2018-September, no. January, pp. 257–261 (2018)
6. Lee, C.M., Narayanan, S.S.: Toward detecting emotions in spoken dialogs. *IEEE Trans. Speech Audio Process.* **13**(2), 293–303 (2005)
7. Banse, R., Scherer, K.R.: Acoustic profiles in vocal emotion expression. *J. Pers. Soc. Psychol.* **70**(3), 614–636 (1996)
8. Hozjan, V., Kačič, Z.: Context-independent multilingual emotion recognition from speech signals. *Int. J. Speech Technol.* **6**(3), 311–320 (2003)
9. Ramakrishnan, S.: Recognition of Emotion from Speech: A Review. *Speech Enhancement, Modeling and Recognition- Algorithms and Applications* (2012)
10. Sebe, N., Cohen, I., Huang, T.S.: Multimodal emotion recognition. *Handbook of Pattern Recognition and Computer Vision*, 3rd Edition (2005)
11. Zhang, Q., Wang, Y., Wang, L., Wang, G.: Research on speech emotion recognition in E-learning by using neural networks method. In: 2007 IEEE International Conference on Control and Automation, ICCA (2007)
12. Jing, S., Mao, X., Chen, L.: Prominence features: effective emotional features for speech emotion recognition. *Digit. Sig. Proc. Rev. J.* **72**(October), 216–231 (2018)
13. Albornoz, E.M., Milone, D.H., Rufiner, H.L.: Spoken emotion recognition using hierarchical classifiers. *Comput. Speech Lang.* **25**(3), 556–570 (2011)
14. Özseven, A., Düğenci, T., Durmuşoğlu, M.: A content analysis of the research approaches in speech emotion. *Int. J. Eng. Sci. Res. Technol.* **7**, 1–27 (2018)
15. Kishore, K.K., Satish, P.K.: Emotion recognition in speech using MFCC and wavelet features. In: Proceedings of the 2013 3rd IEEE International Advance Computing Conference, IACC 2013 (2013)
16. Yousefpour, A., Ibrahim, R., Hamed, H.N.A.: Ordinal-based and frequency-based integration of feature selection methods for sentiment analysis. *Expert Syst. Appl.* **75**, 80–93 (2017). <https://doi.org/10.1016/j.eswa.2017.01.009>
17. Shu, L., et al.: A review of emotion recognition using physiological signals. *Sensors (Switz.)* **18**(7), 2074 (2018)
18. Oosterwijk, S., Lindquist, K.A., Anderson, E., Dautoff, R., Moriguchi, Y., Barrett, L.F.: States of mind: emotions, body feelings, and thoughts share distributed neural networks. *Neuroimage* **62**(3), 2110–2128 (2012). <https://doi.org/10.1016/j.neuroimage.2012.05.079>
19. Pessoa, L.: Emotion and cognition and the amygdala: from “what is it?” to “what’s to be done?” *Neuropsychologia* **48**(12), 3416–3429 (2010). <https://doi.org/10.1016/j.neuropsychologia.2010.06.038>
20. Koolagudi, S.G., Sreenivasa Rao, K.: Emotion recognition from speech: a review. *Int. J. Speech Technol.* **15**(2), 99–117 (2012). <https://doi.org/10.1007/s10772-011-9125-1>
21. Winkielman, P., Niedenthal, P., Wielgosz, J., Eelen, J., Kavanagh, L.C.: Embodiment of cognition and emotion. In: Mikulincer, M., Shaver, P.R., Borgida, E., Bargh, J.A. (eds.) *APA handbook of personality and social psychology, Volume 1: Attitudes and social cognition.*, pp. 151–175. American Psychological Association, Washington (2015). <https://doi.org/10.1037/14341-004>
22. Fernández-Caballero, A., et al.: Smart environment architecture for emotion detection and regulation. *J. Biomed. Inform.* **64**, 55–73 (2016). <https://doi.org/10.1016/j.jbi.2016.09.015>
23. Guan, H., Liu, Z., Wang, L., Dang, J., Yu, R.: Speech emotion recognition considering local dynamic features. In: Fang, Q., Dang, J., Perrier, P., Wei, J., Wang, L., Yan, N. (eds.) *ISSP 2017. LNCS (LNAI)*, vol. 10733, pp. 14–23. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-00126-1_2

24. Cen, L., Wu, F., Yu, Z.L., Hu, F.: A Real-Time Speech Emotion Recognition System and its Application in Online Learning. *Emotions, Technology, Design, and Learning*, Elsevier, Amsterdam (2016)
25. Shuman, V., Scherer, K.R.: Emotions, Psychological Structure of International Encyclopedia of the Social & Behavioral Sciences: Second Edition, Elsevier, Amsterdam (2015)
26. Ekman, P.: 'Basic Emotions'. *Handbook of Cognition and Emotion*, Wiley, Hoboken (2005)
27. Langner, O., Dotsch, R., Bijlstra, G., Wigboldus, D.H.J., Hawk, S.T., van Knippenberg, A.: Presentation and validation of the radboud faces database, *Cognition and Emotion* (2010)
28. Ekman, P.: 'Facial expression and emotion', *American Psychologist* (1993)
29. Bourke, C., Douglas, K., Porter, R.: Processing of facial emotion expression in major depression: a review. *Aust. N. Z. J. Psychiatry* **44**(8), 681–696 (2010)
30. Van den Stock, J., Righart, R., De Gelder, B.: Body expressions influence recognition of emotions in the face and voice. *Emotion* **7**(3), 487–494 (2007). <https://doi.org/10.1037/1528-3542.7.3.487>
31. Banse, R., Scherer, K.R.: Acoustic profiles in vocal emotion expression. *J. Pers. Soc. Psychol.* **70**(3), 614–636 (1996). <https://doi.org/10.1037/0022-3514.70.3.614>
32. Gulzar, T., Singh, A., Sharma, S.: Comparative analysis of LPCC, MFCC and BFCC for the recognition of Hindi words using artificial neural networks. *Int. J. Comput. Appl.* **101**(12), 22–27 (2014)
33. Shrawankar, U., Thakare, V.M.: Techniques for Feature Extraction In Speech Recognition System: A Comparative Study (2013)
34. Haamer, R.E., Rusadze, E., Lsi, I., Ahmed, T., Escalera, S., Anbarjafari, G.: 'Review on Emotion Recognition Databases', *Human-Robot Interaction - Theory and Application* (2018)
35. Lalitha, S., Geyasruti, D., Narayanan, R., Shravani, M.: Emotion detection using MFCC and cepstrum features. *Procedia Comput. Sci.* **70**, 29–35 (2015)
36. Jackson, P., Haq, S.: Surrey Audio-Visual Expressed Emotion (savee) Database. University of Surrey, Guildford, UK (2014)
37. Liu, Z.T., Xie, Q.M., Wu, W.H., Cao, Y., Mei, Y., Mao, J.W.: Speech emotion recognition based on an improved brain emotion learning model. *Neurocomputing* **309**, 145–156 (2018)
38. Ekman, P., et al.: Universals and cultural differences in the judgments of facial expressions of emotion. *J. Pers. Soc. Psychol.* **53**(4), 712–717 (1987). <https://doi.org/10.1037/0022-3514.53.4.712>
39. Zeng, Z., Pantic, M., Roisman, G.I., Huang, T.S.: A survey of affect recognition methods: audio, visual, and spontaneous expressions. *IEEE Trans. Pattern Anal. Mach. Intell.* **31**, 39–58 (2009)
40. Koduru, A., Valiveti, H.B., Budati, A.K.: Feature extraction algorithms to improve the speech emotion recognition rate. *Int. J. Speech Technol.* **23**(1), 45–55 (2020). <https://doi.org/10.1007/s10772-020-09672-4>
41. Kumar, K., Kim, C., Stern, R.M.: Delta-spectral cepstral coefficients for robust speech recognition. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* (2011)
42. Tiwari, V.: MFCC and its applications in speaker recognition. *Int. J. Emerg. Technol.* **1**, 19–22 (2010)
43. Dave, N.: Feature Extraction Methods LPC, PLP and MFCC In Speech Recognition. *Int. J. Adv. Res. Eng. Technol.* **1**, 1–6 (2013)
44. Yankayi, M.: 'Feature Extraction Mel Frequency Cepstral Coefficients (Mfcc)', pp. 1–6 (2016)
45. Ananthakrishnan, S., Narayanan, S.S.: Automatic prosodic event detection using acoustic, lexical, and syntactic evidence. *IEEE Trans. Audio Speech Lang. Process.* **16**, 216–228 (2008)
46. Kinnunen, T., Li, H.: An overview of text-independent speaker recognition: from features to supervectors. *Speech Commun.* **52**(1), 12–40 (2010). <https://doi.org/10.1016/j.specom.2009.08.009>

47. Wang, W.Y., Biadsy, F., Rosenberg, A., Hirschberg, J.: Automatic detection of speaker state: Lexical, prosodic, and phonetic approaches to level-of-interest and intoxication classification. *Comput. Speech Lang.* **27**, 168–189 (2013)
48. Lyons, J.: ‘Mel Frequency Cepstral Coefficient’, Practical Cryptography (2014)
49. Palo, H.K., Chandra, M., Mohanty, M.N.: Recognition of human speech emotion using variants of Mel-frequency cepstral coefficients. In: Konkani, A., Bera, R., Paul, S. (eds.) Advances in Systems, Control and Automation. LNEE, vol. 442, pp. 491–498. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-4762-6_47
50. Yazici, M., Basurra, S., Gaber, M.: Edge machine learning: enabling smart internet of things applications. *Big Data Cogn. Comput.* **2**(3), 26 (2018). <https://doi.org/10.3390/bdcc2030026>
51. Wang, X., Dong, Y., Hakkinen, J., Viikki, O.: ‘Noise robust Chinese speech recognition using feature vector normalization and higher-order cepstral coefficients’ (2002)
52. Davis, S.B., Mermelstein, P.: Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *Readings in Speech Recognition* (1990)
53. Palaz, D., Magimai-Doss, M., Collobert, R.: End-to-end acoustic modeling using convolutional neural networks for HMM-based automatic speech recognition. *Speech Commun.* **108**, 15–32 (2019). <https://doi.org/10.1016/j.specom.2019.01.004>
54. Passricha, V., Aggarwal, R.K.: A comparative analysis of pooling strategies for convolutional neural network based Hindi ASR. *J. Ambient Intell. Humanized Comput.* **11**, 675–691 (2020)
55. Vimala, C., Radha, V.: Suitable feature extraction and speech recognition technique for isolated Tamil spoken words. *Int. J. Comput. Sci. Inf. Technol.* **11**, 675–691 (2014)
56. Dalmiya, C.P., Dharun, V.S., Rajesh, K.P.: An efficient method for Tamil speech recognition using MFCC and DTW for mobile applications. In: 2013 IEEE Conference on Information and Communication Technologies, ICT 2013 (2013)
57. NithyaKalyani, A., Jothilakshmi, S.: ‘Speech Summarization for Tamil Language’. Intelligent Speech Signal Processing (2019)
58. Stevens, S.S., Volkmann, J., Newman, E.B.: A scale for the measurement of the psychological magnitude pitch. *J. Acoust. Soc. Am.* **8**, 208 (1937)
59. Mitrović, D., Zeppelzauer, M., Breiteneder, C.: ‘Features for Content-Based Audio Retrieval’ (2010)
60. Caruana, R., Niculescu-Mizil, A.: An empirical comparison of supervised learning algorithms. In: ACM International Conference Proceeding Series (2006)
61. Kotsiantis, S.B.: ‘Supervised machine learning: A review of classification techniques’, *Informatica* (Ljubljana) (2007)
62. Luckner, M., Topolski, B., Mazurek, M.: Application of XGBoost algorithm in fingerprinting localisation task. In: Saeed, K., Homenda, W., Chaki, R. (eds.) Computer Information Systems and Industrial Management. CISIM 2017. Lecture Notes in Computer Science, vol. 10244. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_57
63. Sutton, O.: ‘Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction’, Introduction to k Nearest Neighbour Classification (2012)
64. Deng, Z., Zhu, X., Cheng, D., Zong, M., Zhang, S.: Efficient k NN classification algorithm for big data. *Neurocomputing* **195**, 143–148 (2016). <https://doi.org/10.1016/j.neucom.2015.08.112>
65. Okfalisa, I., Gazalba, I., Reza, N.G.I.: ‘Comparative analysis of k-nearest neighbor and modified k-nearest neighbor algorithm for data classification’. In: Proceedings - 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering, ICITISEE 2017 (2018)
66. Skuratovskii, R.V.: The timer compression of data and information. In: Proceedings of the 2020 IEEE 3rd International Conference on Data Stream Mining and Processing, DSMP 2020, pp. 455–459 (2020)

67. Skuratovskii, R.V.: employment of minimal generating sets and structure of sylow 2-subgroups alternating groups in block ciphers. In: Bhatia, S., Tiwari, S., Mishra, K., Trivedi, M. (eds.) Advances in Computer Communication and Computational Sciences. Advances in Intelligent Systems and Computing, vol. 759. Springer, Singapore (2019). https://doi.org/10.1007/978-981-13-0341-8_32
68. Romanenko, Y.O.: Place and role of communication in public policy. Actual Probl. Econ. **176**(2), 25–26 (2016)
69. Skuratovskii, R.V., Williams, A.: Irreducible bases and subgroups of a wreath product in applying to diffeomorphism groups acting on the Möbius band. Rend. Circ. Mat. Palermo Ser. **2**, 1–19 (2020). <https://doi.org/10.1007/s12215-020-00514-5>
70. Skuratovskii, R.V.: A method for fast timer coding of texts. Cybern. Syst. Anal. **49**(1), 133–138 (2013)
71. Skuratovskii, R., Osadchyy, V., Osadchyy, Y.: The timer incremental compression of data and information. WSEAS Trans. Math. **19**, 398–406 (2020)
72. Drozd, Y., Skuratovskii, R.V.: Generators and relations for wreath products. Ukr. Math. J. **60**(7), 1168–1171 (2008)
73. Zgurovsky, M.Z., Pankratova, N.D.: System Analysis: Theory and Applications, p. 446. Springer Verlag, Berlin (2007)



A Smart City Hub Based on 5G to Revitalize Assets of the Electrical Infrastructure

Santiago Gil¹(✉), Germán D. Zapata-Madrigal¹, and Rodolfo García Sierra²

¹ Grupo T&T, Facultad de Minas, Universidad Nacional de Colombia, Medellín, Colombia
{sagilar, gdzapata}@unal.edu.co

² Head of Innovation Observatory, Enel-Codensa S.A. E.S.P. (Enel Group), Bogotá, Colombia
rodolfo.garcia@enel.com

Abstract. The electrical infrastructure has certain assets which have been underused, commonly named as dead assets, such as the light, distribution, and transformer poles. This infrastructure can be exploited with short-range 5G base stations, a technology which is expected to arrive soon, as a strategy to implement new 5G IoT networks and new smart city services. This project proposes the development of a smart city Hub which uses the dead electrical infrastructure with the purpose of offering technological services based on 5G. The Hub 5G is intended to integrate applications from different domains and industries, and at the same time, use open data sources to expose integrated services to the community where the Hub is located. The Hub provides a smart digital ecosystem for connected people, data, things, and services, a requirement for nowadays' solutions. This initiative enables the application of technologies for society and the reinforcement of collaborative strategies between different industries and public and private sectors.

Keywords: IoT · 5G · Integrated services · Smart city

1 Introduction

The 5G panorama generates high expectations of new applications, scenarios, capabilities and business models that will emerge due to the fact that it is an interoperable technology across all domains, and that will further drive the digital transformation of all industrial and economic sectors. Additionally, as well as the IP protocol, cellular technologies and 5G networks are convergent technologies on which many other technologies and applications such as IoT, data analysis and Big Data will be based [1]. The connectivity ecosystem between things, devices, machines, humans, and other entities will generate many business opportunities and strategies, and they will be highly dependent on private and secure technology [2]. Those opportunities are oriented for both businesses and consumers; the number of devices connected to the network will grow enormously and the market associated with 5G could grow more than 2.2 trillion dollars in the period 2020 – 2034; in addition, it is expected to generate higher incomes in companies (about 40%) 5 years after the start of 5G. The key to 5G will be 1) transmission

speed, 2) the number of connected devices, which 4G cannot supply completely, and 3) non-public networks over 5G; this will make it possible to improve manufacturing processes in factories, such as, for example, increasing the number of automated robots or vehicles. Other business models based on data may be exploited, and data may be monetized as an asset. On the other hand, consumers will have faster access, better connectivity, and a better disposition of connected resources to be informed, monitor and control what is happening around them and their assets. Some of the most benefited sectors are automotive and mobility, media and content (gaming industry), public sector and smart cities, health, manufacturing, and utilities [3].

The digital transformation, which is extremely necessary for smart -grid, city, industry- applications, requires significant development of wireless communication networks and software resources. The concept of the industrial revolution in which research institutions and companies have focused on is framed under the name of Industry 4.0, an industry based on digital assets and the integration of physical components with virtual components and software [4]. It highlights the importance of the virtualization of data and the implementation of virtual environments which link the physical processes and data with an interoperability strategy, which are in the core of this project.

It is also necessary to consider the coexistence of technologies with 5G and its support in IoT applications for hybrid and interoperable solutions, since the complete ecosystem and the integration of technologies is what consolidates the convergency of the IoT and its exploitation for all the sectors and industries. Finally, IoT technologies behave as enablers one another [5]. It is even important to analyze aspects such as requirements, challenges, and research road maps in this area, to consolidate maturity and exploit the full potential of 5G technology and its supporting technologies, and then this may be the basis for new inventions and developments in the future.

This project proposes a 5G-based smart Hub for smart city services to work along with available electrical infrastructure to extend services and portfolio of energy service providers from the revitalization of unused assets such as light and distribution poles. There are several applications in smart cities and utilities that can be implemented from this solution as are presented here in energy price data, mobility, video surveillance, energy consumption, and other potential applications such as vehicle to grid integration and recommendations to improve efficiency in utility consumption. The approach can act as an enabler for the integration of people, industries, vehicles, and commerce with the smart grid, creating an intelligent and interconnected ecosystem for multi-domain applications.

The remainder of this paper is as follows: Section 2 reviews the state of the art. Section 3 presents the design of the solution. Section 4 shows the configuration of the system and deployment scenario. Section 5 presents the implementation of the system. And finally, Sect. 6 concludes the results and presents some research directions.

2 State of the Art

5G is attributed to an opportunity in the technology sector to drive the digital transformation of all industries, and even with significant social and economic impacts; this with a global perspective of benefits for all regions and people [6]. 5G became a reality from

a concept, although it is only in its early stages [7]. For this, multiple requirements must be addressed, such as wireless communications, energy efficiency of communications, robustness and capacity of networks, business models, data governance, among others. That is why 5G is being a standardized development from organizations and alliances, not just individual inventions of single companies.

The review of Li et al. [1] highlights current research, key technologies, and research trends / challenges for 5G-IoT. This review shows all the heterogeneous technologies required for the development of IoT, such as LoRa, Wi-Fi, ZigBee, Bluetooth, among others. The closest to 5G-based IoT implementation was NB-IoT and LTE-M at the launch of 3GPP Low Power Wide Area Technologies [8]. Now 5G is committed to massifying IoT connections and consolidating concepts such as Industrial IoT or Smart Cities with real-time applications.

As presented in [9], the telecommunication technologies have advanced and have been tailored for their use in the electrical systems. This paper presents various applications and projects that have been carried out for smart metering, control and monitoring of networks, energy management in homes and buildings, management of the electrical grids, integration and monitoring of renewable energy sources, monitoring and low and medium voltage network management, and other projects involving information and communication technologies (ICT), in which the “EPIC-HUB” project is mentioned, a GSM-based system for monitoring consumption energy and the energy produced. The approach of Karpenko et al. in [10] also proposes the use of interoperable IoT ecosystems for the integration of electrical systems. To do this, they have focused on standardized and interoperable communication protocols and their integration with an IoT ecosystem where things, cloud systems and IoT users interact. The application cases they propose are a parking and charging system for intelligent electric vehicles, considering the integration of the ecosystem; it means, including the associated platforms, the mobile application and the semantic integration of the data with the platform and users.

Some of the challenges and research directions that are presented for the development of smart cities are first of all the consolidation of emerging technologies for social welfare, of which Big Data, Machine Learning, the IoT, and the Data analysis are key to achieve this. Then it comes the extension of the applications for different domains to improve energy efficiency, traffic management, parking, mobility as a service and security [11]. In order to integrate applications from different domains, multi-domain and interoperable services are required. The IoT has begun to be widely used for these types of needs, through interoperable services based on IoT technologies and cloud computing, which allow the promotion of new offerings for smart cities [12]. In the article presented in [13], different smart city services are grouped by technological clusters: content management systems, network technologies, data storage and distributed systems, business analytics and intelligence, emerging technologies, and innovation for Smart City. Users' perceptions of the different services proposed by each cluster were then associated to verify the most favorable services and the least favorable (adequate) services. Another proposal to integrate information technology services, and in this case, for the transport infrastructure was carried out in [14], where a City-Hub was used to improve the services, fluidity, comfort, and energy efficiency of the infrastructure of transport networks for smart cities.

The review done by Minoli and Occhiogrosso [15] discusses practical aspects of 5G implementation for IoT applications in smart cities. Among their analysis, the authors place special emphasis on the coexistence of 5G with other IoT technologies such as NB-IoT, 4G or LoRa. It also highlighted the need for small radio cells (micro / Pico antennas) in 5G IoT applications and the importance of signal penetration. The feasibility of implementing microcells in urban poles (light poles) for 5G signal coverage in smart cities is exposed. Among the resources that benefit from 5G IoT there are: smart buildings, transport systems, smart grids, traffic control, pollution monitoring, street lighting management, surveillance, crowdsensing, logistics, smart services, among others.

In the work of Santos et al. [16] a fog computing framework for the autonomous orchestration of functionalities and services in 5G smart cities is proposed. The proposed framework is aimed to improve computing and network performance and provide functionality for data monitoring and analysis. This work refers to other architectures or frameworks for M2M (Machine-to-Machine) communication orchestration and other network technologies, such as ETSI OneM2M, OpenFog Consortium, ETSI MEC (Mobile Edge Computing), ETSI NFV (Network Function Virtualization) NANO, OMA (Open Mobile Alliance) Lightweight M2M, TOSCA (Topology Orchestration Specification for Cloud Applications) and Cloudlet. Additionally, open source tools are mentioned for NFV (OpenStack, OpenBaton, OpenMano), SDN (Software-Defined Networking) (OpenDayLight, ONOS, Ryu) and for M2M (OMA LwM2M, Leshan, 5G-EmPOWER). The Fog computing-based management and orchestration framework consists of 3 layers: sensor layer, fog layer, and cloud layer. The sensor layer is composed of sensors and actuators, connects to the Fog layer via LPWAN (Low Power Wide Area Network) gateways with Fog nodes. Each Fog node is an autonomous system that manages computational resources, and at the same time, these nodes communicate with Cloud nodes, which are responsible for global management and control operations in the network. The Fog and Cloud nodes can communicate one another through a Fog P2P protocol, which allows the exchange of information, and improves the decision-making process related to the provisioning of 5G smart city resources.

Just as the previous article ([16]) refers to standards such as ETSI for the definition of its architectures in communication networks for 5G, it is necessary to consider other proposals, such as the Industrial Internet Reference Architecture (IIRA) of the IIoT (Industrial IoT) Consortium [17] and the Reference Architectural Model Industrie 4.0 (RAMI) of the Plattform Industrie 4.0 [18].

The work of Marques et al. [19] proposes an IoT architecture for the application of waste management in smart cities as a strategy to address the challenge of extra population in cities for the coming years. The proposed architecture considers indoor and outdoor implementation, both in order to automatically separate waste correctly. The MQTT, CoAP and HTTPS protocols were implemented as communication/application protocols on the platform. Thus, the platform incorporates Cyber-Physical Systems (CPS), advanced data communication systems, and embedded intelligence to deal with challenges of smart cities. Additionally, it is composed of four layers: the physical objects layer, the communication layer, the Cloud platform layer and the services layer.

Cheng et al. [20] proposed an architecture for Industrial IoT based on 5G for smart manufacturing applications. An analysis is also made of the eMBB (enhanced Mobile

Broadband), mMTC (Massive Machine Type Communication) or MIoT (Massive IoT) and URLLC (Ultra Reliable Low Latency Communication) scenarios, and technologies and challenges for 5G-based IIoT. The architecture was designed to include various application scenarios in smart manufacturing, including real-time data acquisition from heterogeneous sources in manufacturing plants, collaborative network manufacturing, human-machine interaction, Automated Guided Vehicles (AGV) collaboration, convergence of digital twins between the physical and cyber layers, product design and manufacturing and maintenance based on virtual reality and augmented reality. The architecture also includes advanced digital technologies such as big data, cloud computing, edge computing, digital twin, virtual/augmented reality and service-oriented manufacturing. Considerations in the architecture to comply with service-oriented manufacturing are highlighted as: 1) large-scale intelligent interconnection of heterogeneous equipment, 2) high reliability, high transmission rate and low latency for monitoring and control, 3) edge computing-based IIoT using SDN technology, 4) 3D and multimedia assisted interaction, and 5) low cost and low power consumption in network transmission.

Rahimi et al. [21] also proposed an IoT architecture based on 5G and other advanced digital technologies for the integration of applications, services and generated data. The technologies incorporated in this architecture are: Nanochips, millimeter Wave (mmWave), heterogeneous networks (HetNet), device-to-device (D2D) communication, NFV, SDN, Advanced Spectrum Sharing and Interference Management (Advanced SSIM), Mobile Edge / Cloud Computing, Data Analytics and Big Data. The architecture consists of eight layers: the physical device layer, the communication layer, the Edge (Fog) computing layer, the data storage layer, the service management layer, the application layer, the collaboration and processes layer, and the security layer.

The IoTEP platform proposed in [22] is another sample of open IoT-based systems for energy data management and analysis. This application has a greater emphasis on the energy data analysis layer and its application in buildings, although it has a suitable structural design for integration with other information systems to support external applications. A similar proposal is presented in [23], where emphasis is placed on communications, in this case LPWAN technologies for the implementation of IoT platforms for energy systems. Although this last proposal does not have an open data consideration, it proposes an architecture based on communication technologies and interoperable application protocols to guarantee the integration of data with external applications and highlights the importance of communication technologies in this type of project. In the previous two platforms and in [24], the IoT platforms have several associated applications such as smart metering, distributed energy resources (DER) and demand response management.

Other common initiatives in the framework of Smart Cities refers to data projects propagated through Hubs, which are platform-type devices provided with sufficient communication protocols for the integration of data and services in the IoT ecosystem: devices, machines, people, applications, among others. The proposal presented in [25] exposes the concept of City Hub, an IoT Hub that acts as a portal for infrastructure and for other Hubs. On such device, data from multiple verticals can be integrated - Open Data, transport, traffic, air quality, energy, etc. The City Hub is based on a platform as a service (PaaS) framework, which can provide services to citizens, local companies that offer

other added services, and even large companies or external companies that want to offer new services to citizens. For this, the Hub consists of an architecture that associates the physical devices and the available data to the cloud, with a high level of interoperability and accessibility to ensure the connectivity of multiple sources of information, and allow the different actors involved in the ecosystem to interact with it. The IoT Hub proposed in [26] also considers the integration with consolidated IoT platforms available in the cloud, which facilitate the process of integrating data from IoT systems and, furthermore, guarantee highly scalable conditions important in IoT solutions. The data associated with this Hub was fed through multiple sources of information such as energy and gas meters, current meters, a weather station, a solar generation plant, an air quality station and others, available in the ABB Krakow Research Center, Poland. In the case of [27], a semantic data hub was proposed to improve data integration in machine-to-machine communications (M2M) in non-domain dependent applications. Other semantic tools for data interoperability and integration can be implemented on data hubs, as proposed in [14].

In the review carried out in [28], 23 technological projects for Smart Cities services grouped by IoT & Cloud Computing projects are referenced. The groups are IoT, Cloud Computing & Big Data; Cloud Computing & Big Data; Cloud Computing; and Cloud Computing & Cyber-Physical Systems. Some of the projects referenced in this review are: SmartSantander, Padova Smart City, European Platform for Intelligent Cities (EPIC) Project, Clout, OpenMTC, OpenIoT, Concurrivity, Sentilo, Scallop4SC (SCALable LOgging Platform for Smart City), CiDAP, WindyGrid, SMARTY, U-City, and Civitas. The nature of these projects is open source and addressed to the benefit of citizens.

3 Design of the Solution

3.1 Hub 5G Architecture

Taking previous works as reference, it is very important to consider modular communication architectures structured by layers, which include the necessary components of the solution: service integration, IoT, people, applications, local computing (Edge) and cloud computing, among other considerations.

The communication architecture proposes the coexistence of wireless IoT technologies, the integration of multiple communication protocols and the provision of services to devices of different nature; touch-screen interfaces for use of people in site, mobile apps, web applications, interfaces between machines, devices, and sensors, among others.

The approach towards 5G which is intended to be provided in the solution, and the Hub's application on revitalization of assets and infrastructure of the electrical system, such as light, distribution, and transformer poles, it is possible to consider a network infrastructure composed of 5G microcells with Pico or micro antennas, as considered in [15]. This strategy focuses on generating value from existing infrastructure, and revitalizing assets to provide more services, different from the functionalities that they are oriented to.

Additionally, the core of the application, which will be supported by 5G technology, must provide enough interoperability, both horizontally between wireless communication devices and technologies, and vertically to satisfy the interaction between different

domains regarding smart cities and their integration with other sectors such as smart grids and transportation systems.

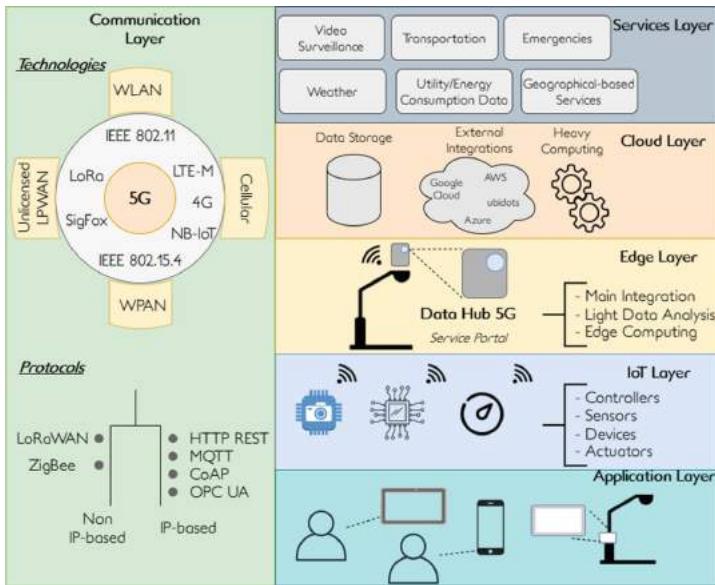


Fig. 1. Communication architecture of the hub 5G.

The architecture is presented in Fig. 1. The proposed architecture is made up of six layers: the application layer, where users can access through available devices or tablets on the poles where the Hub 5G is located to access all the information and services that the Hub provides. The IoT layer, where it is performed the connection between the IoT network and the Hub 5G; the idea is to connect cameras for artificial vision, air quality and meteorology sensors, energy, gas and water meters, and other IoT-enabled devices that allow data to be acquired and services to be offered to the community. The Edge layer is where the 5G data Hub is located. This layer is in charge of integrating all the layers in the system, and then, offering the services as a portal of services and benefits for people and things. The 5G data Hub will be in charge of processing light data, and thus, lighten the data transmission to the cloud and decrease response times taking advantage of edge computing. The Cloud layer offers integration with other platforms that allow storing, processing, and managing system resources and data at a massification level. This layer is essential because of the nature of offering smart city services, where it is extremely important to have applications that third parties can provide; the system is also prone to working with big data, so opting for heavy cloud processing is much more efficient and faster. Light processing can be done on the Edge layer. The services layer is responsible for orchestrating all the functionalities that people and things will be able to access through the Hub 5G and its integrations, including citizen security services through video surveillance, information of public transport and mobility, emergencies and catastrophes, suggestions based on geographic

location such as restaurants, shopping centers, etc., weather information, health, and utility consumption. Finally, the communication layer is transparent to all the layers of the system and provides communication capabilities through IoT technologies and protocols to all components of the network. A greater emphasis is placed on Low Power Wide Area Networks (LPWANs), especially cellular LPWAN technologies, to be used within the system which are intended to coexist with 5G. In the same order of ideas, it is planned to consolidate a robust core based on IoT technologies and protocols which will be ready for the arrival of 5G in smart city applications.

3.2 Data Integration Model

Controlling the scenarios of data integration in the Hub 5G is extremely important to both things and people; it means, the interaction with the environment, actors, actions, and perceptions. Therefore, it is necessary to consider a model which includes all the possible interactions that the Hub can have. Moreover, it is required to establish the model of the application through which the services are going to be offered to users and things.

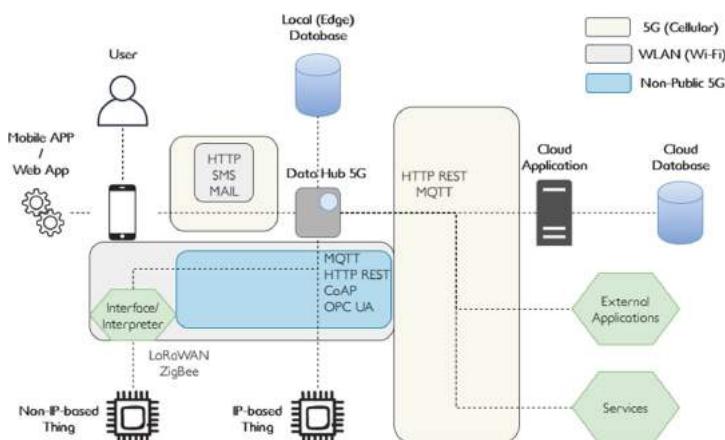


Fig. 2. Data integration model of the hub 5G.

Figure 2 show the data integration architectural model which considers how the links between the different actors of the system will be implemented. This model presents the protocols and technologies to be used in each of the communication stages and how the link is implemented for data storage for Edge - few data, specific information - and for Cloud - storage of all events and data generated in the system -. The communication between devices is oriented to be using Wireless Local Area Networks with Wi-Fi (IEEE 802.11) and through private 5G networks (Non-Public networks) according to Release 16 of 3GPP that will allow to do such implementations. Several protocols are provided so different types of devices can be integrated to the Hub 5G ecosystem. The communication between the Hub 5G (Edge) and the Cloud will be done by means of 5G through the

integration of MQTT and HTTP services. The external services are handled using HTTP Application Program Interfaces (APIs). Finally, the integration of the mobile application and end users is via WLAN (Wi-Fi), Short Message Service (SMS), or Email via 5G interfaces. HTTP is used as the main protocol for the integration of services of the Hub 5G, the mobile application, and the external services.

4 Configuration of the System

The configuration of the Hub 5G environment is carried out using a Raspberry PI 3 or a Coral Edge TPU; both are similar in terms of performance and interfacing options, the former with more options to be adapted to several peripherals, and the later with better performance to process streaming video while recognizing. The better option is to work with both boards to achieve better performance. In the case the image recognition is carried out outside a TPU, the idea is to only deal with the Raspberry PI 3, because its capacities in terms of RAM and CPU are better than the Coral Edge. The correct strategy is to let the Raspberry PI 3 to orchestrate all the services and peripherals and let the Coral Edge TPU only to infer streaming video, making a powerful Hub 5G ecosystem. The integration between both boards is seamless because both are based on Debian distributions; the Raspberry PI runs a Raspbian OS 10 and the Coral Edge TPU runs a Mendel Day 4.0 OS.

Regarding to the physical configuration, the Hub 5G system depends on 4 components: the core (Raspberry PI and/or Coral Edge TPU); the camera, which is intended to run inside an aerial vehicle (drone) but currently runs using a PI or USB Camera; the cellular modem to enable the 4.5G and later the 5G interface when available; and finally, a smart tablet attached to the Hub 5G to allow the users to access smart city services in light poles.

4.1 Mobile Network Initialization

The Hub 5G communication interface is achieved via the cellular NB-IoT, LTE-M, and 4G modem U-Blox Sara R410M. The communication is achieved with two options: native cellular AT commands and native use of communication protocols and using a dial-up manager to establish internet connection with the modem. Both options are great to meet the communication requirements of the Hub 5G, but the dial-up option is more flexible as it allows a transparent connection to the internet while using as many protocols over IP as you need. Again, the best option is to use a combination of both alternatives, sometimes using dial-up and sometimes using AT commands.

The Sara R410 modem offers a variety of options to send/receive messages using IoT communication alternatives. For example, the most common, the SMS is carried out through the “AT + CMGS” command; the HTTP REST protocol is achieved through a combination of “AT + UHTTP”, “AT + UHTTPC”, and “AT + URDFILE” commands, available for GET, POST, PUT, and DELETE requests; the MQTT publish/subscribe protocol is achieved through a combination of “AT + UMQTT” and “AT + UMQTTC” commands.

The dial-up configuration is done by means of the “wvdial” Debian service. It acts as a network manager for the cellular module and provides a network interface and the IP address. The required parameters to establish the dial-up are as follows:

```
[Dialer Defaults]
Modem = /dev/ttyUSB1
Modem Type = Analog Modem
Baud = 115200
ISDN = 0
Stupid mode = 1
Idle Seconds = 0
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0
Init3 = AT+CGDCONT=1,"IP","MOBILE_OPERATOR_APN"
Phone = *99***1#
Username = "USER"
Password = "PASSWORD"
New PPPD = yes
Dial Command = ATD
```

After initialized the wvdial service, a ppp interface is then assigned to the Hub 5G with an IP provided by the network operator.

4.2 Application Initialization

Once all the components connected to the Hub 5G are ready, the Hub application server is carried out. There are two server applications: 1) the application for face recognition with browser streaming output and 2) the back-end application where data is processed, and services are orchestrated for the Hub 5G and the smart city services. Additionally, there is a client application which runs on a mobile device, a Tablet located along with the Hub 5G infrastructure in a light pole.

Face Recognition Application

The face recognition application is done by means of a pre-trained face detection model with an own database of faces selected for testing. It is encouraged to include faces of the community in the database to know whether an unknown is in the surroundings and keep track the surveillance of people who are near the Hub 5G. The face recognition model is based on the FaceNet model and the Histogram of Oriented Gradients (HOG) method to make it lighter and be able to run on a Raspberry PI. In contrast, Convolutional Neural Networks (CNN) methods are too heavy for them to run on the Raspberry because of its processing capabilities.

Hub 5G Back-End Application

The back-end application is developed using the Python-Django framework. It manages the resources of the Hub to provide data to the mobile application, Internet access and native AT commands, and it also manages the available data in the system to perform data analysis. The native AT commands are managed from this application through serial communication between the Hub 5G and the U-Blox cellular modem.

This application is also in charge of accessing the open government datasets to map the smart city services of the Hub with updated open data. This initiative takes benefits

from the following open data portals (they all are in Spanish as the official language of the country):

- MEData of the Medellín City Government. (<http://metadata.gov.co/>).
- Colombia ESRI Open Data (<http://datosabiertos.esri.co/>).
- Open Data of the Metropolitan Area of Medellín (<https://datosabiertos.metropol.gov.co/>).
- Colombia Open Data (<https://www.datos.gov.co/>).
- SIATA portal, weather management of the Metropolitan Area of Medellín (https://siata.gov.co/siata_nuevo/).

Additionally, the application also requests an open API which returns a 5-min feed energy price data which serves as a base for scheduling energy demand based on an hourly tariff. The API is available on <https://hourlypricing.comed.com/api?type=5minutefeed>.

All these features have been implemented with the purpose of providing smart city services for transport, utilities, health, safety, weather, and leisure.

An additional approach that was covered in this application regarding the Coronavirus pandemic, was the incorporation of daily updated data of the virus in the country. It offers an additional service which for this precise moment becomes essential in a way to provide information of the current situation, which would help the community to be aware of self-care actions.

Mobile Application

The mobile application is developed using the native Android method with Android Studio. It is intended to offer the visual component of the Hub 5G services. The mobile application is oriented as the graphical user interface of the Hub 5G, from it the resources, data, and requests of the application are accessed in order to provide smart city services in a simple and technological way to the community. The mobile app is in the application layer of the Hub 5G architecture and orchestrates the services of the service layer. The mobile application has at least one activity per offered service, i.e., one or more activities for video surveillance, transportation, emergencies, weather, utility data, and geographical-based services. This application takes advantage of available open data platforms as mentioned above in Sect. 3.2 to implement some of the stated services. To accomplish such features, some activities implement WebView and Google Maps Activities to satisfy the requirements of some services, for example, the mobility service.

4.3 Deployment Scenario

When the massification of 5G is available for the seamless deployment in the cities, the desired implementation scenario is expected to cover the locations where the energy service providers have dead electrical infrastructure such as transformer, light, or distribution poles with several 5G Hubs. Each Hub will be provided with a pico-antenna to cover at least 100m² around. This is a strategy not only to offer smart city services to the community, but also to extend the service portfolio of the energy service providers from

the already deployed electrical infrastructure. This way, the energy service providers can offer 1) the electricity service, 2) value-added information to interested third-parties, 3) provide welfare to their customers, and 4) establish strategic alliances between technology enterprises, such as network service providers, water service providers, public and private transport, etc. The proposed scenario is currently not too viable, because of how the 2G, 3G, and 4G antennas are designed for, but with the arrival of the different types of cells of 5G, it would become more feasible; for example, using Femto (max 32 users) or Pico (max 128 users) cells, distributed along the distribution tower in the city, a powerful integrated smart city scenario can be achieved.

Now, the 5G is just in its first phases, so, it is necessary to work with the available technology. That's why this project centers the communication interface in a cellular modem with NB-IoT and LTE-M, which are technologies that are expected to work along with the 5G and are scalable to future 5G releases in terms of IoT connectivity. The Hub 5G currently operates as an end user in a cellular network, using the U-Blox modems, but in a further stage, it is expected to change these end nodes (modems) to Femto or Pico antennas when available in the local market. It exposes and strategy to be completely prepared for the 5G arrival and the arrangement of technological services based on 5G for smart cities and energy service providers.

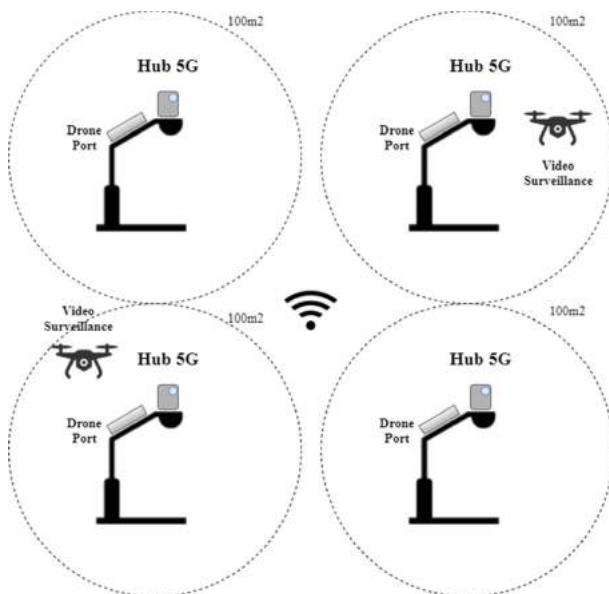


Fig. 3. Hub 5G system expected scenario.

The Hubs are intended to be interconnected among them, each one with a drone port and several drones sending video streaming to perform the video surveillance in the location automatically. Then, a collaborative and interconnected system of Hubs will be achieved to provide smart city services over several locations, just as shown in Fig. 3. For now, this project carries out the prototyping phase of this technological development.

5 Implementation of the Hub

The Hub 5G prototype is intended to be assembled into a light pole in Universidad Nacional de Colombia at Campus Robledo (Facultad de Minas). This prototype implementation allows the analysis for a subsequent stage of mass deployment in smart cities.

The procedure of deployment and verification of the Hub is as follows: 1) The hardware and communication components are integrated together and then initialized. 2) The software which manages the drivers, services, and communication interfaces is initialized. 3) The back-end application for accessing the services is launched. 4) The mobile application is initialized. 5) All the peripherals of the system are checked.

5.1 Hardware and Communication

The Hub is initialized by powering up the Raspberry PI and peripherals, then, the communication protocols begin to work. The background services are handled using the Crontab service. The main module to deal with the communication via cellular technology is the U-Blox Sara R410M module; after its initialization and the corresponding service, it should return a ppp interface such as follows:

```
ppp0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 10.140.164.4 netmask 255.255.255.255 destination 10.64.64.64
        ppp txqueuelen 3 (Point-to-Point Protocol)
        RX packets 5 bytes 62 (62.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 6 bytes 101 (101.0 B)
        TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Once the cellular interface is ready, a bridge between the wireless Wi-Fi interface and the cellular module is configured to set the WAP in the Hub.

Additional protocols or interfaces, such as LoRaWAN could be handled with several approaches. In the case it was decided to include the Hub 5G as a LoRaWAN end node, it is achieved by implementing a serial interface between the Hub 5G (Raspberry PI) and a Microchip RN2903 LoRa module embedded in a LoRa 2 Click MikroE module. It could be done through the Raspberry PI native pins or using a FTDI USB to serial converter. Another approach for doing this is going further to the application layer of the OSI model and integrate the LoRaWAN protocol via MQTT. It enables the cooperation between protocol and at the same time, the tracking of LoRaWAN data from the Hub. The integration via MQTT using the SARA module can be using native AT commands or using Mosquitto or any MQTT library directly from the application with the ppp interface.

The native AT commands for integrating MQTT are as follows:

```

AT+UMQTT=0,"111" # identifier
AT+UMQTT=1,1883 ### port
AT+UMQTT=2,"serverDNS" # server
#AT+UMQTT=3,"IP",1883 # IP with port (alternative)
AT+UMQTT=4,"user","pass" # user / password
AT+UMQTT=10,36000 # timeout
#AT+UMQTT=11,1,2 # secure option
#AT+UMQTT=12,1 # clean session
AT+UMQTTC=1 # login
AT+UMQTTC=2,0,0,"topic/subtopic","message" # publish to a topic
AT+UMQTTC=4,0,"application/+device/+/rx" # subscribe to a topic
AT+UMQTTC=6 # read message
# Example response
+UMQTTC: 6,1
OK
+UUMQTTCM: 6,1
Topic:application/1/device/d7d9aaebd65709ce/rx
Msg: LoRaWAN message codified as JSON object.

```

Similarly, when using the Python alternative, it is done through the paho-mqtt library and its methods. The implementation of other communication protocols such as OPC UA and CoAP can be seamlessly integrated into the Hub via Python modules when necessary. Good options to implement these protocols are for example: Python-OPCUA (FreeOPCUA) and CoAPthon.

5.2 Integration Between Back End and Mobile Application

Once the configuration of the Hub 5G is done, it is the time to go on with the applications. As said before, the software components of the ecosystem consist of the main back-end application, which remains on the Hub controller itself, the integration of external application and services on the cloud, and the mobile application which acts as the Graphical User Interface. At this time, no applications on the cloud are developed on our own, so, they cannot be administrated for specific purposes, but it is expected to be done soon.

The back-end application is designed using a Model-View-Controller pattern, which most of the views are oriented to the integration with the mobile application; it means that the back-end application has views without any HTML layout because they only transfer data or specific images to the mobile application. The back-end application is also in charge of mapping the external application or services to the mobile application and the services provided by the Hub. As stated before, this initiative intends to reuse as much as possible open web-based platforms which are employed for the benefit of the region and the community.

The resources of the application are organized in a strategic way according to the services it is intended to offer. Other resources are gathered directly from its public resource to the mobile application, such as the weather service which is available on https://siata.gov.co/siata_nuevo/. The updated mobility in the place is acquired using the *setTrafficEnabled* method on the Google Maps activity. Table 1 provides an overview of the resources provided by the back-end application. Once the resources are available on the back-end application in the local network, they are accessed in the mobile application using HTTP requests with Volley or integrated directly with Web Views.

Table 1. Resources provided by the hub 5G back-end application.

Resource URI	Resource type	Description
Health/coronavirus-department	API	An API which maps the data of the departments (states) of Colombia with updated new confirmed and total cases
Health/coronavirus-department-view	Map View	A resource which maps the departments (states) coronavirus data into a Folium map to be displayed on the web and on the mobile application via Android Studio WebView
Health/coronavirus-city-view	Map View	A resource which maps the cities coronavirus data into a Folium map to be displayed on the web and on the mobile application via Android Studio WebView
Health/coronavirus-dailycases	API	An API which maps the daily data of coronavirus in Colombia with current and new cases, deaths, and recovered people
Health/coronavirus-dailycases-view	Graph View	A resource which maps the daily cases API to charts to be displayed on the web and on the mobile app using Plotly
Transport/publictransportroutes	API	An API which maps open data of public bus and bus-metro integration routes in Medellín. This data is then mapped into a Google Map View on the mobile application
Transport/bikeroutes	API	An API which maps open data of available bike routes in Medellín. This data is then mapped into a Google Map View on the mobile application
Utilities/energypriceAPI	API	An API which maps the Comed open API for 5-min feed (https://hourlypricing.comed.com/api?type=5minutefeed)
Utilities/watertariff	API	An API which maps open water tariff data of the local water service provider

(continued)

Table 1. (continued)

Resource URI	Resource type	Description
Utilities/watertariffview	Graph View	A resource which maps the water tariff API to charts to be displayed on the web and on the mobile application
Utilities/energytariff	API	An API which maps open energy tariff data of the local energy service provider
Utilities/energytariffview	Graph View	A resource which maps the energy tariff API to charts to be displayed on the web and on the mobile application
Utilities/gastariff	API	An API which maps open gas tariff data of the local gas service provider
Utilities/gastariffview	Graph View	A resource which maps the gas tariff API to charts to be displayed on the web and on the mobile application
Emergencies/emergencyAPI	API	An API to receive requests from the mobile application for different types of emergencies

5.3 Integrated Ecosystem

The resources are then available to be accessed from the android application in a smartphone or tablet which will be located on a light pole with the Hub for the people to use it. The app has a main view, which has a button for testing, for services, and for configuration. The main functionality is on the services menu, which is composed of several services for mobility, utilities, health, and so on. Some of the services have sub-menus to access different data or functionalities. Then, when entering each final view of the application, there is an informative visualization to provide useful data to people who navigate in the app. This is a strategy to bring people to the processes involved in the energy and electrical systems and to be more confident with digitization. All these components bring the community to be a pillar in the transformation of smart cities and smart grids, such as stated by Lu in its framework of interoperability of Industry 4.0 [4]. Some of the services provided by the application are as follows:

- Services which show information about the tariff for the different utilities for different hours or months. These services are oriented to inform people how to employ the utilities consciously in order to reduce utility billing. Figure 4 show an overview of these services in the mobile application. The energy price API view uses a 5 min feed for 24 h, the water and gas tariffs views use the tariff data of the local water/gas service provider in Medellín discriminated by socioeconomic stratum, and the energy tariff view, which is not shown in the figure, uses the tariff data of the local energy service provider discriminated by hourly tariffs.

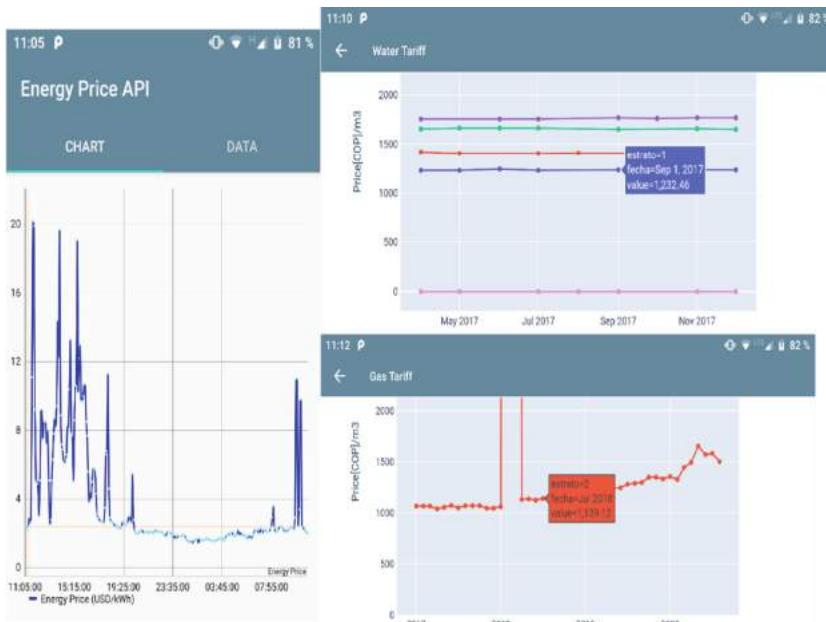


Fig. 4. Utility services provided by the hub 5G. on the left, the energy price API. on the right, above: the water tariff view, below: gas tariff view.

- Services which show information about the updated traffic nearby, the available bike routes in the city, and the available bus and bus to metro public transport routes. the tariff for the different utilities for different hours or months. These services are oriented to inform people of local mobility in order to promote sustainable mobility and the use of bikes and public transport. Figure 5 show an overview of these services in the mobile application. On the left, the updated mobility view uses the Google services to plot in a green to red scale how much traffic is on a route. On the middle, the bike routes view plots the available bike routes in the city to contribute to a safe biking. On the right, the public transport routes view plots the available bus transport routes in the city to promote the use of public transport instead of private.

- Services which show information about the updated coronavirus state in Colombia. These services are oriented to inform people of the condition of this critical situation discriminated by department (state) and city in order to promote more care regarding the virus. Figure 6 show an overview of these services in the mobile application. The Coronavirus department map view plots the current state of every department in Colombia in a yellow to dark red scale according to the criticality of each region. It also shows the specific location of the Hub with a blue marker. The Coronavirus daily update view shows the new confirmed, recovered, and dead cases every day up to the last entry, which is updated by the government daily.

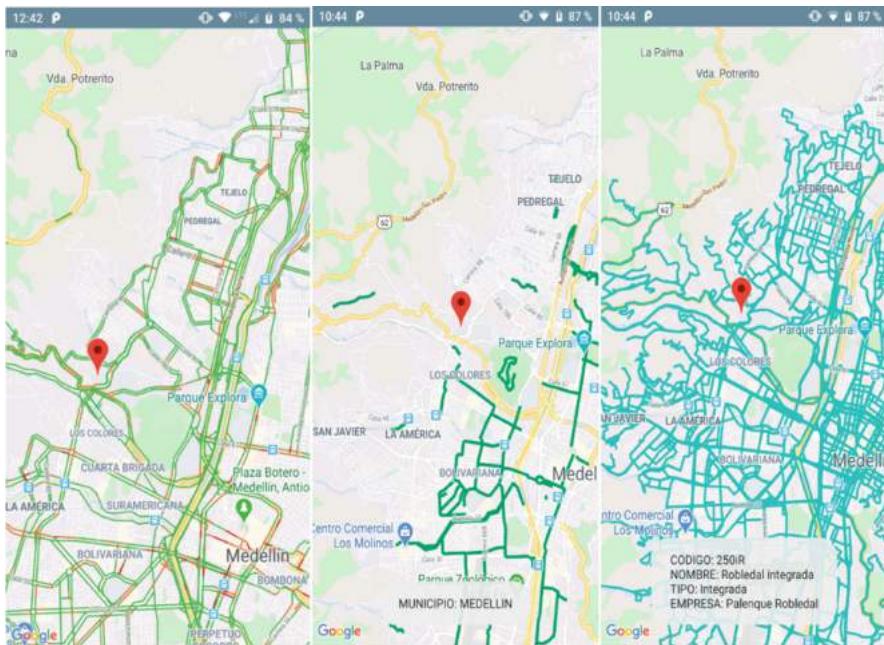


Fig. 5. Mobility services provided by the hub 5G mobile app. from left to right: updated mobility, bike routes, and public transport routes.

- Services which provide support in case of risky situations nearby. These services are oriented to allow people the request of emergencies because of any abnormal situation such as fire, earthquake, landslide, river overflow, etc. nearby. It provides a video surveillance feature to track people who attend the place, if they belong to the community or are unknown. With the proper database, it can identify criminals walking around the zone. Additionally, it also provides the emergency view, which offers a form to fill in and submit in case of any emergency such as fire, earthquake, or any other emergency nearby.

Additional views such as the Weather view points to the responsive site of SIATA, a public agency which administrates weather and meteorological measurements in Medellín and its Metropolitan Area. On the other hand, the geographical-based service uses the Nearby Search Query API of Google Places API to map and mark leisure and entertainment places close to the Hub, such as restaurants, malls, and cinemas.

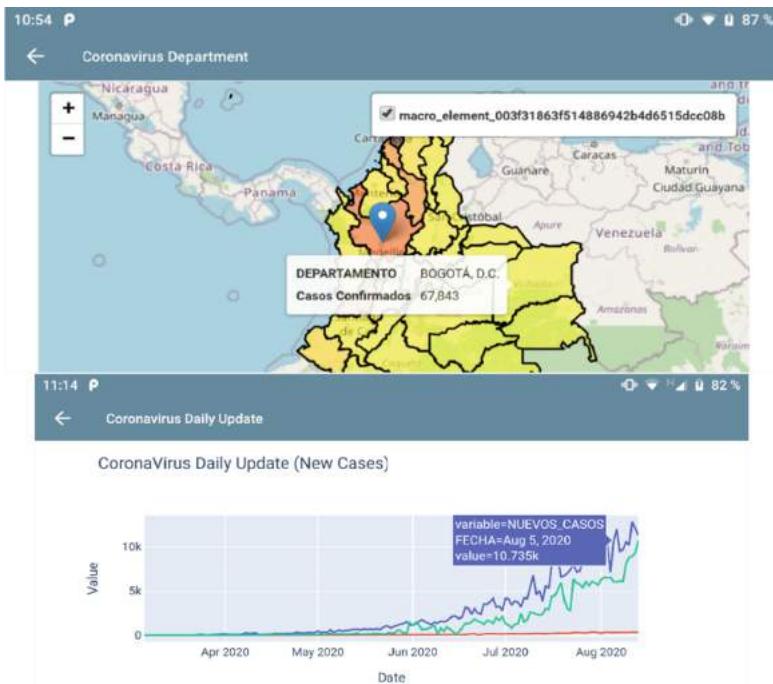


Fig. 6. Coronavirus service provided by the hub 5G mobile app. above: discriminated confirmed cases by department (state). below: daily update with new cases.

6 Conclusions and Future Work

In this application it was addressed an initiative to revitalize the dead electrical infrastructure such as light and transformer poles by implementing a technological approach to converge smart city services taking the electrical industry as a main pillar of the digital transformation of cities and society.

With the deployment of data-based electrical systems and their integration with other domains and industries, this approach becomes very convenient for the energy service providers to offer new technological services and expand the service portfolio to new areas different from just providing energy, using already deployed infrastructure. It can also lead to cooperation between companies from different domains to improve the local market income through a collaborative performance and strategic alliances in the 5G IoT ecosystem. A good example for that would be the cooperation between the energy service provider of a city, the network service operator, and the early alert (emergencies) department; the energy provider serves with the infrastructure to deploy a 5G micro-cell network composed of several cells, the network operator provides the cellular base stations, and the early alert department would be alert to any emergency or catastrophe which would happen in any zone within the cellular network and could respond to it faster. In the same order of ideas, the energy provider would be able to link the smart meters covered by the network, the network operator would offer the cellular

service normally with this network, and the early alert department could increase the amount of emergency/catastrophe data to implement predicting models.

This contribution exposes a first adoption of 5G technology for short-range base stations but considering its massification to deploy a long-range 5G IoT network. It is important consider the arrival of 5G as a reality in the years to come, and then, be prepared to work along with this technology becomes essential to implement new technological services for the new concept of cellular communication.

It is also relevant to highlight how novel technologies, applied and oriented correctly to the population, can assist to a better understanding of the processes, the news, and the current situation. The technological initiatives of this era, the era of the Industry 4.0, smart cities, smart grids, cyber-physical systems, and so on, should consider the incorporation of the social factor as a main point to state the purposes, then, a convergent digital ecosystem for the benefit of society, environment, and economy would be addressed.

In next steps of this project, it is expected to cover some components that were not included this time. Just to mention some of the future steps: first, embedding the video surveillance with a drone video streaming network to automate the face recognition and alert in the place. Another one would be the integration of the Hub 5G approach with smart homes; it would enable active interaction between homes' utility measurements, emergencies, billing, and even strategic marketing when appropriate. A similar approach would be the integration of the network of Hubs to vehicle, especially electric vehicles, as a way to implement vehicle-2-grid and grid-2-vehicle communication, and then, implement strategies for managing energy demand and propose energy-efficient plans in electric vehicles. Finally, the services which are provided related to safety, i.e., video surveillance, emergencies, and even health, could be then integrated with a public agency to improve the response time and data for treatment of such critical situations.

References

1. Li, S., Xu, L.D., Zhao, S.: 5G internet of things: a survey. *J. Ind. Inf. Integr.* **10**, 1–9 (2018). <https://doi.org/10.1016/j.jii.2018.01.005>
2. Palattella, M.R., et al.: Internet of things in the 5g era: enabling technologies and business models. *IEEE J. Sel. Areas Commun.* **34**, 510–527 (2016)
3. GSMA: Internet of THIngs in the 5G Era - Opportunities and Benefits for Enterprises and Consumers. 26 (2019)
4. Lu, Y.: Industry 4.0: a survey on technologies, applications and open research issues. *J. Ind. Inf. Integr.* **6**, 1–10 (2017). <https://doi.org/10.1016/j.jii.2017.04.005>
5. Barakabitze, A.A., Ahmad, A., Mijumbi, R., Hines, A.: 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. *Comput. Netw.* **167**, 106984 (2020). <https://doi.org/10.1016/j.comnet.2019.106984>
6. Díaz de Terán, L.M.: 5G, la oportunidad que Europa no puede dejar escapar. https://www.abc.es/tecnologia/informatica/soluciones/abci-luis-manuel-diaz-teran-opportunidad-europa-no-puede-dejar-escapar-202003050129_noticia.html. Accessed 06 March 2020
7. Pye, A.: 5G from concept to reality. <https://www.environmentalengineering.org.uk/news/5g-from-concept-to-reality-2590/>. Accessed 06 March 2020
8. GSMA: 3GPP Low Power Wide Area Technologies (LPWA). GSMA White Pap, pp. 19–29 (2016)

9. Andreadou, N., Guardiola, M.O., Fulli, G.: Telecommunication technologies for smart grid projects with focus on smart metering applications. *Energies* **9**(5), 375 (2016). <https://doi.org/10.3390/en9050375>
10. Karpenko, A., et al.: Data exchange interoperability in IoT ecosystem for smart parking and EV charging. *Sensors* (Switzerland), p. 18 (2018). <https://doi.org/10.3390/s18124404>
11. Nambiar, R., Shroff, R., Handy, S.: Smart cities: challenges and opportunities. In: 2018 10th International Conference on Communication Systems & Networks (COMSNETS), COMSNETS 2018. January 2018, pp. 243–250 (2018). <https://doi.org/10.1109/COMSNETS.2018.8328204>
12. Taherkordi, A., Zahid, F., Verginadis, Y., Horn, G.: Future cloud systems design: challenges and research directions. *IEEE Access*. **6**, 74120–74150 (2018). <https://doi.org/10.1109/ACCESS.2018.2883149>
13. Lytras, M.D., Visvizi, A., Sarirete, A.: Clustering smart city services: perceptions, expectations, responses. *Sustain.* **11**, 1–19 (2019). <https://doi.org/10.3390/su11061669>
14. Heddebaut, O., Di Ciommo, F.: City-hubs for smarter cities. the case of lille “EuraFlandres” interchange. *Eur. Transp. Res. Rev.* **10**(1), 1–14 (2017). <https://doi.org/10.1007/s12544-017-0283-3>
15. Minoli, D., Occhiogrosso, B.: Practical aspects for the integration of 5G networks and IoT applications in smart cities environments. *Wirel. Commun. Mob. Comput.* **2019**, 30 (2019). <https://doi.org/10.1155/2019/5710834>
16. Santos, J., Wauters, T., Volckaert, B., de Turck, F.: Fog computing: enabling the management and orchestration of smart city applications in 5G networks. *Entropy* **20**, 4 (2018). <https://doi.org/10.3390/e2001004>
17. Lin, S.-W., et al.: The industrial internet of things volume G1: reference architecture. *Industrial Internet Consortium* 1.80, pp. 1–7 (2017)
18. Plattform Industrie 4.0: Reference Architectural Model Industrie 4.0 (RAMI 4.0) - An Introduction (2016)
19. Marques, P., et al.: An IoT-based smart cities infrastructure architecture applied to a waste management scenario. *Ad Hoc Netw.* **87**, 200–208 (2019). <https://doi.org/10.1016/j.adhoc.2018.12.009>
20. Cheng, J., Chen, W., Tao, F., Lin, C.L.: Industrial IoT in 5G environment towards smart manufacturing. *J. Ind. Inf. Integr.* **10**, 10–19 (2018). <https://doi.org/10.1016/j.jii.2018.04.001>
21. Rahimi, H., Zibaeenejad, A., Safavi, A.A.: A novel IoT architecture based on 5G-IoT and next generation technologies. In: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON) 2018, pp 81–88 (2019). <https://doi.org/10.1109/IEMCON.2018.8614777>
22. Terroso-Saenz, F., González-Vidal, A., Ramallo-González, A.P., Skarmeta, A.F.: An open IoT platform for the management and analysis of energy data. *Futur. Gener. Comput. Syst.* **92**, 1066–1079 (2019). <https://doi.org/10.1016/j.future.2017.08.046>
23. Song, Y., Lin, J., Tang, M., Dong, S.: An internet of energy things based on wireless LPWAN. *Engineering* **3**, 460–466 (2017). <https://doi.org/10.1016/J.ENG.2017.04.011>
24. Kuzlu, M., Rahman, M.M., Pipattanasomporn, M., Rahman, S.: Internet-based communication platform for residential DR programmes. *IET Netw.* **6**, 25–31 (2017). <https://doi.org/10.1049/iet-net.2016.0040>
25. Lea, R., Blackstock, M.: City hub: a cloud-based IoT platform for smart cities. In: Proceedings of the International Conference on Cloud Computer Technology and Science CloudCom. February 2015, pp. 799–804 (2015). <https://doi.org/10.1109/CloudCom.2014.65>
26. Bajer, M.: Building an IoT data hub with elasticsearch, Logstash and Kibana. In: Proceedings of the 2017 5th International Conference on Future Internet Things Cloud Work. W-FiCloud 2017. January 2017, pp. 63–68 (2017). <https://doi.org/10.1109/FiCloudW.2017.101>

27. Gyrard, A., Bonnet, C., Boudaoud, K.: Enrich machine-to-machine data with semantic web technologies for cross-domain applications. In: 2014 IEEE World Forum Internet Things, WF-IoT 2014, pp. 559–564 (2014). <https://doi.org/10.1109/WF-IoT.2014.6803229>
28. Santana, E.F.Z., Chaves, A.P., Gerosa, M.A., Kon, F., Milojicic, D.S.: Software platforms for smart cities: concepts, requirements, challenges, and a unified reference architecture . ACM Comput. Surv. (CSUR) **50**(6), 1–37 (2016)



LoRa RSSI Based Outdoor Localization in an Urban Area Using Random Neural Networks

Winfred Ingabire^{1,2(✉)}, Hadi Larijani¹, and Ryan M. Gibson¹

¹ School of Engineering and Built Environment, Glasgow Caledonian University, Glasgow, UK

{Winfred.Ingabire,H.Larijani,Ryan.Gibson}@gcu.ac.uk

² Department of Electrical and Electronics Engineering, University of Rwanda-College of Science and Technology, Kigali, Rwanda

Abstract. The concept of the Internet of Things (IoT) has led to the interconnection of a significant number of devices and has impacted several applications in smart cities' development. Localization is widely done using Global Positioning System (GPS). However, with large scale wireless sensor networks, GPS is limited by its high-power consumption and more hardware cost required. An energy-efficient localization system of wireless sensor nodes, especially in outdoor urban environments, is a research challenge with limited investigation. In this paper, an energy-efficient end device localization model based on LoRa Received Signal Strength Indicator (RSSI) is developed using Random Neural Networks (RNN). Various RNN architectures are used to evaluate the proposed model's performance by applying different learning rates on real RSSI LoRa measurements collected in the urban area of Glasgow City. The proposed model is used to predict the 2D Cartesian position coordinates with a minimum mean localization error of 0.39 m.

Keywords: IoT · LoRaWAN · RSSI · Localization · RNN

1 Introduction

IoT connected devices are projected to increase exponentially [1], and this is expected to trigger different smart applications in various domains such as healthcare, agriculture, transportation among others. Low Power Wide Area Networks (LPWAN) such as Long-Range Wide-Area Networks (LoRaWAN) are potential reliable low power connectivity solutions to large-scale IoT networks [2]. Moreover, location-aware applications such as Remote Health Monitoring (RHM) is one of the promising smart applications of LoRaWAN [3]. Additionally, an accurate localization system for sensor nodes is crucial, particularly in applications where locating end devices is critical. A typical outdoor localization method in LoRa networks is the Time Difference of Arrival (TDoA) technique. However, this method performs well in open environments and poorly in

harsh urban areas. In comparison, WiFi and Bluetooth RSSI fingerprint based techniques have been successfully applied for indoor localization [4]. While, end device fingerprint localization based on LoRaWAN RSSI characteristics is a research topic yet to be thoroughly investigated, both in indoor and outdoor environments.

Furthermore, RNN has recently been used to create several resilient models with significant results [23]. Currently, no one has reported RNN algorithms applied in end device localization models for IoT wireless sensor nodes. In this paper, we use real test LoRaWAN RSSI data collected in the urban area of Glasgow City to develop a localization model based on LoRaWAN RSSI characteristics using RNN.

End device localization is vital in location-aware IoT applications whereby a sensor node needs to be accurately located for emergency or maintenance services. RSSI fingerprint localization methods involve mapping RSSI values to corresponding X, Y Cartesian 2D position coordinates [31]. Moreover, RSSI fingerprint localization approaches have been proposed for accurate localization models in the literature using GNSS, Bluetooth, and WiFi wireless technologies and successfully implemented but all with significant drawbacks and limitations. In Addition, the following challenges are among the issues limiting the performance of existing RSSI fingerprint techniques for effective end device localization:

1. Fingerprint data maps are affected by changing environments.
2. Strong fingerprint databases are needed.
3. Manpower is needed for creating fingerprint databases.
4. Non-linearity challenge between end nodes and target gateways.
5. High power-consumption and cost.
6. Complicated infrastructure layout.

Studies using LPWAN for sensor node localization are present in literature, however, frequently limited to specific settings such as a small area or fixed end nodes. Furthermore, no other work has used RNN for LoRaWAN based node localization models (to the best of our knowledge). Therefore, this work aims at using RNN algorithms to develop a resilient novel system to improve location prediction accuracy compared to the existing RSSI fingerprint approaches. The main contributions of this paper are summarized as:

- Development of novel RNN based models for accurate LoRa end device localization.
- Real test data sets collected in Glasgow's urban city are used to train and test the developed localization model with a different number of hidden neurons.
- The performance of the developed RNN based localization model is evaluated, and improved accuracy is achieved by training and testing various RNN network architectures using different learning rates.

This paper is organized as follows: Sect. 2 introduces LoRa, and LoRaWAN. Section 3 discusses the related work. Section 4 gives the details of the methodology and the procedures used. Section 5 presents the obtained results with the

performance analysis of the developed model. Finally, Sect. 6 gives conclusions and future work directions.

2 LoRa and LoRaWAN

The invention of low power wide-area network technologies like LoRaWAN, and other potential network enablers for the dense IoT networks, came as a perfect solution for large scale IoT smart applications. LoRa is a physical layer with the Chirp Spread Spectrum (CSS) modulation technique operated by Semetech within the license-free spectrum, whereas LoRaWAN is the connection protocol stack of the wide-area network of LoRa architecture on the MAC layer [5]. A LoRaWAN network is made up of LoRa end devices connected in a star topology that sends information to one or multiple LoRaWAN gateways. Therefore, the gateways that send the received message to a network server along with a recorded unique message's metadata information, as shown in Fig. 1. Thus, this metadata information enables localization in LoRaWAN networks, whereby gateways serve as anchor points to determine position coordinates. TDoA algorithms use timestamps at which different gateways receive the same message, whereas the most critical metric for Fingerprinting is RSSI. Also, the more gateways that can receive the same message, the more accurate a localization algorithm is in any of the methods [7]. Furthermore, the Spreading Factor (SF) is a LoRa critical parameter that affects localization accuracy, whereby lower SF limits the transmission range of LoRa devices. Consequently, fewer gateways receive the same message [6].

3 Related Work

This section presents an overview of existing RSSI Fingerprint localization techniques, Random Neural Networks, and Gradient Descent Algorithm (GDA).

3.1 Overview of RSSI Fingerprint Localization Methods

Different algorithms using RSSI fingerprinting localization exist in literature using various wireless technologies. However, existing studies frequently investigated mostly indoor environments due to the training phase's many data. Furthermore, limited studies are available for localization analysis in outdoor environments, mainly due to the hard work involved in collecting data. WiFi has been used widely in fingerprint localization [8–10], and [11] whereby a smartphone can be used to record its RSSI and estimate its location using the web. The authors make a comparative performance analysis of different wireless technologies based on RSSI localization in [12]. Research studies using LoRaWAN in fingerprint localization are also available in literature [13–16]. Moreover, Choi. W et al. in [17] used LoRa for positioning using three fingerprint algorithms and confirmed LoRa to be effective with the average accuracy of 28.8 m for the three

algorithms. Aernouts. M et al. in [18] presented fingerprint localization data sets for LoRaWAN and Sigfox in large urban and rural areas with a mean estimation error of 398.40 m. Similarly, the authors in [19–21] also investigated LoRa RSSI based fingerprint localization algorithms with significant estimations.

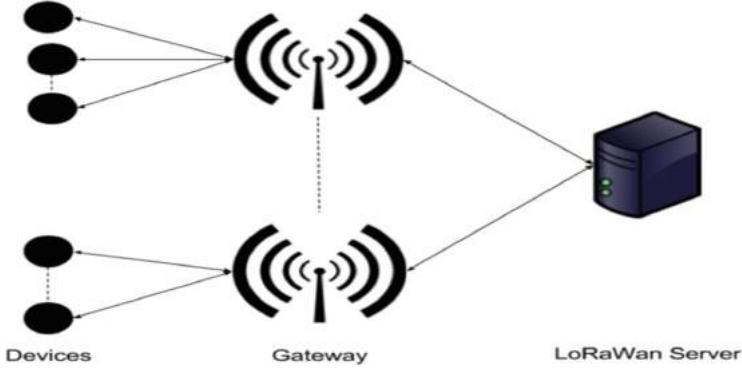


Fig. 1. LoRaWAN network architecture [6]

3.2 Random Neural Networks

RNN has recently been used to create resilient models with significant results. Application in Heating, Ventilation, and Air Conditioning (HVAC) systems by Javed. A et al. in [22–24], energy prediction of non-occupied buildings by Ahmad. J et al. in [25], communication systems, image classification, processing, pattern recognition by Simonyan. K and Zisserman [34], whereas intrusion detection systems were developed and analyzed by Ul-Haq Qureshi. A et al. in [26–28]. Nonetheless, no one has reported RNN algorithms applied in end device localization models (to the best of our knowledge).

RNN is a unique class of ANN developed by Gelenbe [29], consisting of directed N multiple layers of connected neurons that exchange information signals as impulses, using a potential positive (+1) for excitation and a negative (-1) for inhibition signals to the receiving neuron. The potential of each neuron i at time t is represented by a nonnegative integer $K_i(t)$. The neuron i is in excited state if $K_i(t) > 0$ and it is in idle state if $K_i(t) = 0$. If neuron i is excited, it forwards an impulse signal to another neuron j at the Poisson rate r_i . The forwarded signal can reach neuron j as an impulse signal in excitation or inhibition with probabilities $p^+(i, j)$ or $p^-(i, j)$ respectively. Furthermore, the transmitted signal can leave the network with a probability mathematically defined in [25] by the following formulas:

$$c(i) + \sum_{j=1}^N p^+(i, j) + p^-(i, j) = 1, \forall i, \quad (1)$$

$$w^+(i, j) = r_i p^+(i, j) \geq 0, \quad (2)$$

Equally

$$w^-(i, j) = r_i p^- + (i, j) \geq 0. \quad (3)$$

Combining Eqs. 1, 2 and 3

$$r(i) = (1 - c(i))^{-1} \sum_{j=1}^N [w^+(i, j) + w^-(i, j)] \quad (4)$$

The transmission rate between neurons in Eq. 4 is $r(i)$, and is defined as $r(i) = \sum_{j=1}^N [w^+(i, j) + w^-(i, j)]$. While “ w ” determines the matrices of weight updates from neurons, it is always positive as it is a product of transmission rates and probabilities.

In RNN based models, if a signal arrives at neuron (i) with a positive potential, it is denoted by Poisson rate $\Lambda(i)$, whereas a signal with a negative potential arrives at Poisson rate $\lambda(i)$. Therefore, for every node “ i ” the output activation function for that neuron is given by:

$$q(i) = \frac{\lambda^+(i)}{r(i) + \lambda^-(i)}, \quad (5)$$

whereby

$$\lambda^+(i) = \sum_{j=1}^n q(j)r(j)p^+(j, i) + \Lambda(i), \quad (6)$$

with

$$\lambda^-(i) = \sum_{j=1}^n q(j)r(j)p^-(j, i) + \lambda(i). \quad (7)$$

More details about RNN are in [25].

3.3 Gradient Descent Algorithm

Gradient Descent (GD) is a first-order iterative optimization algorithm widely used for training by different researchers. It is used to minimize the cost function whereby the error cost function is given by:

$$E_p = \frac{1}{2} \sum_{i=1}^n \gamma_i (q_j^p - q_j^p)^2, \gamma_i \geq 0 \quad (8)$$

whereby $\gamma \in (0, 1)$ presents the state of output neuron i , likewise q_j^p is a real differential function whereas q_j^p is the predicted output value. As, per Eq. 8, to find the local minima and reduce the error value of the error cost function, the relation between neurons y and z is considered, whereby weights $w^+(y, z)$ and $w^-(y, z)$ are updated by:

$$w_{y,z}^{+t} = w_{y,z}^{+(t-1)} - \eta \sum_{i=1}^n \gamma_i (q_j^p - y_j^p) [\partial q_i \partial w_{y,z}^+]^{t-1}, \quad (9)$$

also:

$$w_{y,z}^{-t} = w_{y,z}^{-(t-1)} - \eta \sum_{i=1}^n \gamma_i (q_j^p - y_j^p) [\partial q_i \partial w_{y,z}^-]^{t-1}. \quad (10)$$

The proposed RNN-LoRa RSSI based localization model is trained using GD, and the calculated weights and biases are updated to the neurons as the algorithm computes the error. More details about GD are found in [25].

4 Methodology

This section gives details about all the procedures used in data collection and developing our RNN-LoRa RSSI based localization model.

4.1 Real Test Measurements

Real test LoRa RSSI datasets were collected in Glasgow City for the training and testing of our developed RNN based end device localization models. Three LoRa SX1301-enabled Kerlink gateways were used to receive the same messages from a MultiTech systems' LoRaWAN mDot end device controlled by Raspberry Pi. The three gateways were at 30m on George More building in Glasgow Caledonian University, at 27m on James Weir building at the University of Strathclyde, and at 25m on top of Skypark building, respectively. The LoRa mote was used to collect data and transmit it to the three gateways simultaneously at a walking speed away from the gateways from different locations. Figure 2 shows a Bing map with three LoRaWAN gateways, LoRa end device locations, and the path is taken for measurements. More details about the procedure used in our measurements are given in [30].

Data Normalization. Our dataset's RSSI absolute values are large, which results in an unstable network due to large weights. Therefore, we scaled our dataset using the Min-Max Normalisation data pre-processing technique to the range of 0 to 1, using the following formula:

$$x_i = \frac{RSSI_i - \min(RSSI)}{\max(RSSI) - \min(RSSI)}, \quad (11)$$

where $RSSI = (RSSI_1, \dots, RSSI_n)$ is the raw RSSI input values and $x(i)$ is the resultant normalized data.

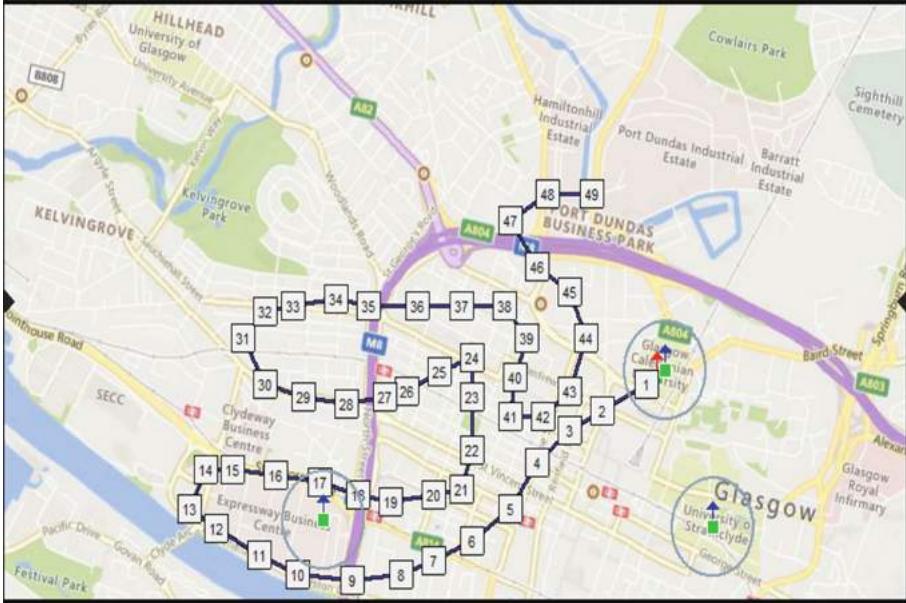


Fig. 2. Glasgow city Bing Map showing three LoRaWAN gateways, LoRa end device positions and the path taken for data collection

4.2 RNN-LoRa RSSI Based Localization Model

End device localization using the RNN approach consists of directed multiple layers of connected nodes in a network. This network is used to develop a model that accurately maps the input to the output using historical data, and the model can then be used to output any desired unknown output. We used the RNN network to develop our proposed model that accurately maps the input RSSI values to the output Cartesian X, Y coordinates using real data measurements collected before, as presented in Fig. 3. Then, the model is used to output any desired unknown position with minimum localization error. RNN learning GD algorithm that provides supervised training is considered for this research study. The RNN is trained to locate each end device in the network service area or grid, and then the trained network is extended further to predict the location of any other sensor nodes on the same network grid based on end-device LoRaWAN-RSSI fingerprints.

Using RNN, we develop RNN-LoRa RSSI-based Localization Model in MATLAB using fingerprint data set created from real measurements taken in Glasgow City with three LoRaWAN gateways. The dataset contains for every message transmitted by the moving end device, the RSSI value at each of the three gateways that received the message with ground position coordinates of the end device at every transmission. Furthermore, we recorded -200 as the RSSI value for every gateway that did not receive a particular message.

5 Results and Performance Analysis

In this section, various RNN architectures for a number of hidden neurons of 6, 12, and 18 were modelled and analysed with 0.01, 0.1, and 0.4 learning rates. Corresponding training and test results of mean localization errors are recorded. A minimum mean localization error of 0.39 m is obtained in a sample size of 1931 data points, whereby 80% of the dataset was used for training and 20% for testing the model with a minimum training mean square value of 0.005 m. The performance of the developed RNN based localization model is analysed using the average localization error (AE) defined as follows:

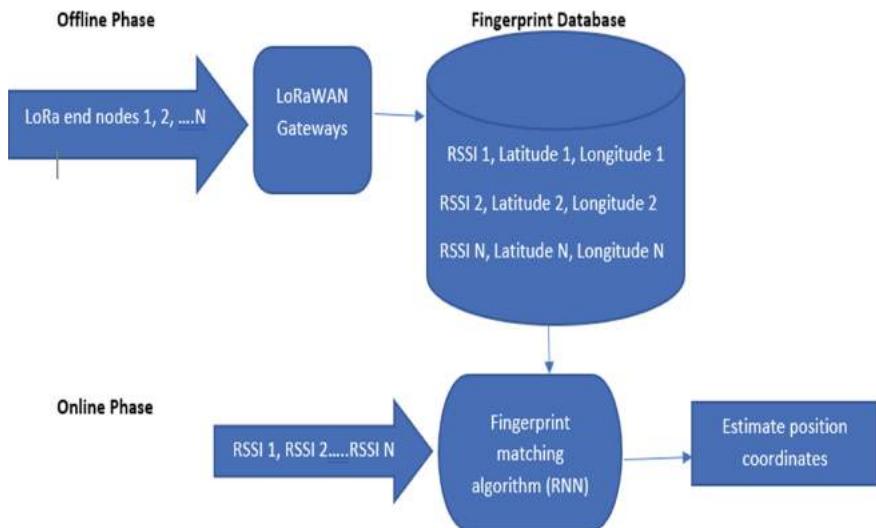


Fig. 3. RSSI fingerprint localization approach

$$AE = \sum_{i=1}^n ((X_{real} - X_{pred})^2 + (Y_{real} - Y_{pred})^2)^{0.5} \quad (12)$$

whereby (X_{real}, Y_{real}) is the real true pre-recorded position using GPS and (X_{pred}, Y_{pred}) is the predicted position of unknown location estimated by the LoRa RSSI based localization system developed using RNN. The total number of samples used in our localization dataset is given by n. Figure 4 presents results for mean localization error values for all the three RNN-LoRa RSSI based localization system architectures using 0.01, 0.1, and 0.4 learning rates. The results show that for the first system network architecture (RNN-1); increasing the learning rate from 0.01 to 0.1 improved localization performance by minimizing the mean localization error. However, increasing the learning rate further to 0.4

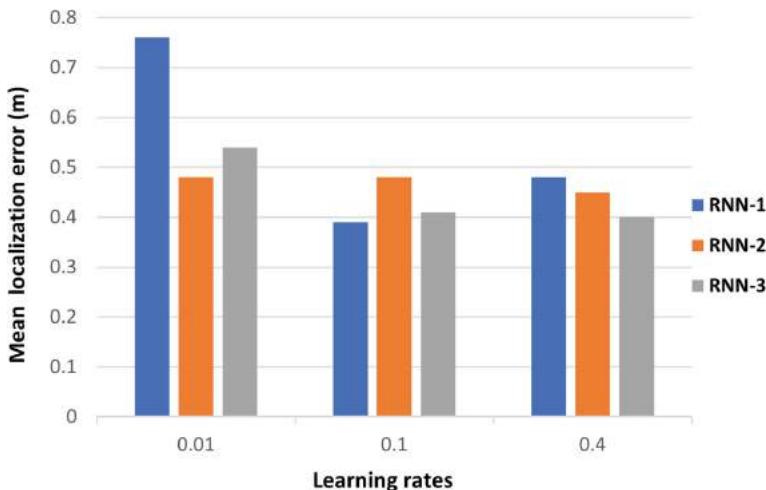


Fig. 4. Performance analysis between RNN based localization network architectures with different learning rates

led to an increase in the mean localization error. Hence, using 0.1 learning rate had the best performance, whereby only six hidden neurons were used.

For the second system architecture (RNN-2), increasing the learning rate from 0.01 to 0.1 increased the system's mean localization error whereas increasing the learning rate further to 0.4 decreased the error. Hence, using 0.01 learning rate had the best performance where twelve hidden neurons were used.

Likewise, for the third system architecture (RNN-3); eighteen hidden neurons were used, and the learning rate increased with the decrease in the mean localization error. Furthermore, three input neurons and two output neurons are considered for all the three RNN localization model architectures in our analysis.

In general, our results show that the developed RNN LoRa RSSI based localization model performed best with a learning rate of 0.01 and when twelve hidden neurons were used. Also, increasing the number of hidden neurons does not significantly improve the performance of the system apart from when the highest learning rate of 0.4 was used.

A comparative performance analysis of localization accuracy of all analysed RNN based localization architectures with different learning rates is summarised in Table 1.

According to our results, the developed RNN based localization models are reliable with significant minimum mean localization errors and outperform the conventional LoRa localization approaches reported by different researchers in literature [7, 18, 31, 32], and [33] with high accuracy for outdoor localization services.

Table 1. Performance of Different RNN based Localization Architectures

Learning rates	Mean localization error (m)		
	RNN-1	RNN-2	RNN-3
0.01	0.76	0.39	0.48
0.1	0.48	0.48	0.45
0.4	0.54	0.41	0.40

6 Conclusion and Future Work

In this study, we presented a LoRa RSSI based outdoor localization system using Random Neural Networks. Three RNN architectures are used to evaluate our localization model's performance using data collected in the urban area of Glasgow City, considering 0.01, 0.1, and 0.4 as learning rates. The proposed model performed best with a minimum mean localization error of 0.39 m. Moreover, the analyzed RNN based localization system architectures showed that increasing the number of hidden neurons does not improve the localization system's performance unless a high learning rate is used. Our results are significant with high-level accuracy for outdoor positioning in urban areas and give important insights for using LoRaWAN and RNN in localization systems. We plan to do a comparative performance analysis of our developed localization model to existing localization models found in literature.

Acknowledgment. This work was funded by the Commonwealth Scholarships in the UK in partnership with the Government of Rwanda.

References

1. Statista Says a Five-Fold Increase in Ten years in Internet-Connected Devices by 2025 Will Significantly Increase the Internet's Promise of Making the World Connected. <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>. Accessed 27 October 2020
2. Ahmad, A.I., Ray, B., Chowdhury, M.: Performance evaluation of loraWAN for mission-critical IOT networks. Commun. Comput. Inf. Sci. CCIS **1113**, 37–51 (2019)
3. Kharel, J., Reda, H.T., Shin, S.Y.: Fog Computing-Based Smart Health Monitoring System Deploying LoRa Wireless Communication. IETE Tech. Rev. (Inst. Electron. Telecommun. Eng. India) **36**(1), 69–82 (2019)
4. Poulose, A., Han, D.S.: UWB indoor localization using deep learning LSTM networks. Appl. Sci. **10**(18), 6290 (2020)
5. Semtech, “LoRaWANspecificationv1.1.” <https://www.lora-alliance.org/technology>. Accessed 27 October 2020
6. Ghosly, S.: “All about LoRa and LoRaWAN” . <https://www.sghosly.com>
7. Bissett, D.: “Analysing tdoa localisation in LoRa networks”. Delft University of Technology (2018)

8. Zghair, N.A.K., Croock, M.S., Taresh, A.A.R.: Indoor localization system using Wi-Fi technology. *Iraqi J. Comput. Commun. Control Syst. Eng.* **19**(2), 69–77 (2019)
9. Hernández, N., Ocaña, M., Alonso, J.M., Kim, E.: Continuous space estimation: increasing wifi-based indoor localization resolution without increasing the site-survey effort. *Sensors (Switzerland)* **17**, 147 (2017)
10. Janssen, T., Weyn, M., Berkvens, R.: Localization in low power wide area networks using Wi-Fi fingerprints. *Appl. Sci.* **7**(9), 936 (2017)
11. Zafari, F., Gkelias, A., Leung, K.K.: A survey of indoor localization systems and technologies. *IEEE Commun. Surv. Tutor.* **21**(3), 2568–2599 (2019)
12. Sadowski, S., Spachos, P.: RSSI-based indoor localization with the internet of things. *IEEE Access* **6**, 30149–30161 (2018)
13. Kwasme, H., Ekin, S.: RSSI-based localization using LoRaWAN technology. *IEEE Access* **7**, 99856–99866 (2019)
14. Anjum, M., Khan, M.A., Hassan, S.A., Mahmood, A., Gidlund, M.: Analysis of RSSI fingerprinting in LoRa networks. In: 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) IWCMC 2019, no. June, pp. 1178–1183 (2019)
15. Lin, Y.C., Sun, C.C., Huang, K.T.: RSSI measurement with channel model estimating for IoT wide range localization using LoRa Communication. In: Proceedings of the 2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS) ISPACS 2019, pp. 5–6 (2019)
16. Goldoni, E., Prando, L., Vizziello, A., Savazzi, P., Gamba, P.: Experimental data set analysis of RSSI-based indoor and outdoor localization in LoRa networks. *Internet Technol. Lett.* **2**(1), e75 (2019)
17. Choi, W., Chang, Y.S., Jung, Y., Song, J.: Low-power LoRa signal-based outdoor positioning using fingerprint algorithm. *ISPRS Int. J. Geo Inf.* **7**(11), 1–15 (2018)
18. Aernouts, M., Berkvens, R., Van Vlaenderen, K., Weyn, M.: Sigfox and LoRaWAN datasets for fingerprint localization in large urban and rural areas. *Data* **3**(2), 1–15 (2018)
19. Lam, K.H., Cheung, C.C., Lee, W.C.: LoRa-based localization systems for noisy outdoor environment. In: International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), vol. 2017-Octob, pp. 278–284 (2017)
20. ElSabaa, A.A., Ward, M., Wu, W.: Hybrid localization techniques in LoRa-based WSN. In: ICAC 2019–2019 25th IEEE International Conference on Automation and Computing (ICAC), no. September, pp. 1–5 (2019)
21. Lam, K.H., Cheung, C.C., Lee, W.C.: RSSI-based LoRa localization systems for large-scale indoor and outdoor environments. *IEEE Trans. Veh. Technol.* **68**(12), 11778–11791 (2019)
22. Javed, A., Larijani, H., Ahmadinia, A., Emmanuel, R., Mannion, M., Gibson, D.: Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for HVAC. *IEEE Internet Things J.* **4**(2), 393–403 (2017)
23. Javed, A., Larijani, H., Wixted, A., Emmanuel, R.: Random neural networks based cognitive controller for HVAC in non-domestic building using LoRa. In: Proceedings of the 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC) 2017, pp. 220–226 (2017)
24. Javed, A., Larijani, H., Ahmadinia, A., Gibson, D.: Smart random neural network controller for HVAC using cloud computing technology. *IEEE Trans. Ind. Inform.* **13**(1), 351–360 (2017)

25. Ahmad, J., Larijani, H., Emmanuel, R., Mannion, M., Javed, A., Phillipson, M.: Energy demand prediction through novel random neural network predictor for large non-domestic buildings. In: Proceedings of the 11th Annual IEEE International Systems Conference (SysCon) 2017, pp. 1–6 (2017)
26. Qureshi, A.U.H., Larijani, H., Javed, A., Mtetwa, N., Ahmad, J.: Intrusion detection using swarm intelligence. In: 2019 UK/China Emerging Technologies (UCET) , pp. 1–5 (2019)
27. Qureshi, A.U.H., Larijani, H., Ahmad, J., Mtetwa, N.: A novel random neural network based approach for intrusion detection systems. In: Proceedings of the 2018 10th Computer Science and Electronic Engineering (CEEC) 2018, pp. 50–55 (2019)
28. Qureshi, A.U.H., Larijani, H., Mtetwa, N., Javed, A., Ahmad, J.: RNN-ABC: a new swarm optimization based technique for anomaly detection. *Computers* **8**(3), 59 (2019)
29. Gelenbe, E.: Random neural networks with negative and positive signal and product form solution. *Neural Comput.* **1**(4), 502–510 (1989)
30. Wixted, A.J., Kinnaird, P., Larijani, H., Tait, A., Ahmadiania, A., Strachan, N.: Evaluation of LoRa and LoRaWAN for wireless sensor networks. In: Proceedings of the IEEE Sensors, pp. 5–7 (2017)
31. Anagnostopoulos, G.G., Kalousis, A.: A Reproducible Comparison of RSSI Fingerprinting Localization Methods Using LoRaWAN (2019)
32. Purohit, J.N., Wang, X.: “LoRa based Localization using Deep Learning Techniques,” p. 2019, “California State University” (2019)
33. Nguyen, T.A.: “LoRa Localisation in Cities with Neural Networks,” p. 71, “Delft University of Technology” (2019)
34. Simonyan, K., Zisserman, A.: “Very deep convolutional networks for large-scale image recognition”. In: International Conference on Learning Representations (ICLR) ICLR 2015 - Conference on Track Proceedings, pp. 1–14 (2015)



A Deep Convolutional Neural Network Approach for Plant Leaf Segmentation and Disease Classification in Smart Agriculture

Ilias Masmoudi^(✉) and Rachid Lghoul

Al Akhawayn University in Ifrane, Hassan II Avenue, 53000 Ifrane, Morocco
I.Masmoudi@awi.ma

Abstract. This paper concerns an approach for developing a set of tools based on computer vision and deep learning to detect and classify plant diseases in smart agriculture. The main reason that motivated this work is that early detection of plant diseases can help farmers effectively monitor the health of their culture, as well as make the best decision to avoid the spread of the pathogens. In this work, a novel way for training and building a fast and extensible solution to detect plant diseases with images and a convolutional neural network is described. The development of this methodology is achieved in two main steps. The first one introduces Mask R-CNN to give bounding boxes and masks over the area of plant leaves in images. The model is trained on the PlantDoc dataset made of labeled images of leaves with their corresponding bounding boxes. And the second one presents a convolutional neural network that returns the class of the plant. This CNN is trained on the PlantVillage dataset to recognize 38 classes across 14 plant species. Experimental results of the proposed approach show an average accuracy of 76% for leaf segmentation and 83% for disease classification.

Keywords: Computer vision · Convolutional neural network · Segmentation · Classification

1 Introduction

An early detection of plant disease is crucial to keep a crop in a healthy condition, but their identification can be a tedious task in many parts of the world due to the unavailability of the required equipment [1]. Moreover, it has been reported that diseases and pests can reduce yields by more than 50%, and small farmers are the most vulnerable to this issue [2].

For efficient crop management, the correct identification of the disease should be fast and cheap. With the huge leap in technological advancement in the past few years, high-resolution cameras, smartphones, and computers have become widespread and accessible to a large portion of the population [1]. All these factors can make an automated solution feasible especially that deep learning techniques have shown their ability to perform exceptionally well on complex tasks.

It has been stated that between 60% and 70% of plant diseases are detected from leaves [3]. Early detection of plant diseases can help farmers effectively monitor the health of their culture, as well as make the best decision to avoid the spread of the pathogens.

Many methods are currently used to detect plant diseases. Lab methods can include Polymerase Chain Reaction (PCR), immunofluorescence (IF), or flow cytometry (FCM) [4]. However, most of these techniques are very costly, time-consuming, and require field expertise. Consequently, researchers have tried to find cost-efficient and reliable techniques to help agriculture advancement. In [5], the authors used manual feature extraction combined with canny edge detection and did histogram matching to test the health of the plant. Another use of computer vision in plant disease detection is shown in [6], where the authors used CIELAB, YCbCr, and HSI color spaces to segment disease spots. With the rapid advancement of AI, many machine learning techniques were proposed as well. The authors of [7] proposed an approach based on K-means as a clustering step and a neural network for disease classification. More recently, deep learning approaches that take advantage of large volumes of data and faster machines made their appearance. In [8], the authors applied a deep convolutional neural network to detect 13 different diseases from plant leaves. And in [9], the authors built a dataset called PlantDoc to demonstrate the feasibility of building a scalable and cost-effective solution using computer vision. The authors also believed that segmenting leaves from images can enhance the utility of their dataset. Mask R-CNN can be used for both, leaf detection and segmentation. However, training the model on multiple classes requires more training time; training data needs to have bounding boxes and is not very scalable. For example, if the farmer decides to add a new plant, the upper layers of the R-CNN model would need to be retrained to include the new leaves. On the other hand, having a model that generalized to detect every leaf would result in a much quicker process. Since only the second model would need to be retrained.

For these reasons, a deep convolutional neural network approach using Mask R-CNN will be covered for both plant leaf segmentation and a second CNN model for disease classification.

The primary step is to detect and segment leaves from the images, and Mask-RCNN is used because of its impressive results in similar tasks [10]. The model is trained on a dataset called “PlantDoc” and is made up of 2,598 images. A second convolutional neural network is trained using the “Plant Village” dataset and then added on top for the leaf segmentation model. The following section explains in detail the methodology, and Sect. 3 shows the results of training the two models.

2 Material and Methods

2.1 Mask -RCNN

The state-of-the-art segmentation model is Mask RCNN [10]. The architecture is modeled in two parts:

The first part is composed of a backbone model like Resnet, VGG, or Inception. Then, the output is fed to a region proposal convolutional neural network that in its turn returns the zones containing objects in the feature maps [10]. In the second part, the

outputs are fed to an ROI align because the fully connected layers are of fixed size. The first fully connected layer is used to predict classes with a SoftMax activation function and the second is a regressor that returns bounding boxes around the objects. The ROI aligned feature maps are also fed to a convolutional neural network and return the masks over every ROI [11].

2.2 Convolutional Neural Networks

In recent years, deep learning techniques demonstrated their ability to make the best use of large volumes of data. Given the complexity of leaf images, convolutional neural networks have the upper hand since their architecture allows learning complex patterns without the need to build handcrafted features [12].

Convolution is a special operation that allows for feature extraction, in which a kernel is applied to the input with an element-wise product at every location and their sum is output at each given position. The output tensor is called a feature map. This operation is repeated for every kernel and the set of feature maps represents the characteristics detected from the input matrix. The main hyperparameters of convolution are the size and number of kernels [12]. And, it is a common practice to use 32 or 64 kernels of size 3×3 for simple tasks.

2.3 Data Augmentation

For certain tasks, the available dataset is not large enough, and this could lead to the underfitting of the model. Luckily, many techniques have been developed and used in the literature to increase the size of the dataset to learn the most important features and achieve better generalization [13, 16]. Pawara, et al. presented in [13] some useful augmentation parameters. During training, we could use one or a combination of parameters by specifying a range or a value to each one of them. Then, the images are randomly generated over those specified limits.

2.4 Leaf Segmentation

The initial step is to detect and segment the leaves from the images. The main reason for this is that a plant can have multiple diseases at the same time and we would like to remove the background to reduce noise during classification.

For the PlantDoc dataset, instead of training the Mask-RCNN model for every class, we considered all images as a single class for two main reasons: training on multiple classes for localization and disease classification simultaneously required more training time. Besides, using two different models; one for leaf detection and the other for disease classification gives more flexibility in terms of maintenance. For instance, if we need to add a new class, we would only have to train the disease classification model.

Figure 1 shows a sample image from the PlantDoc dataset with its corresponding bounding boxes.

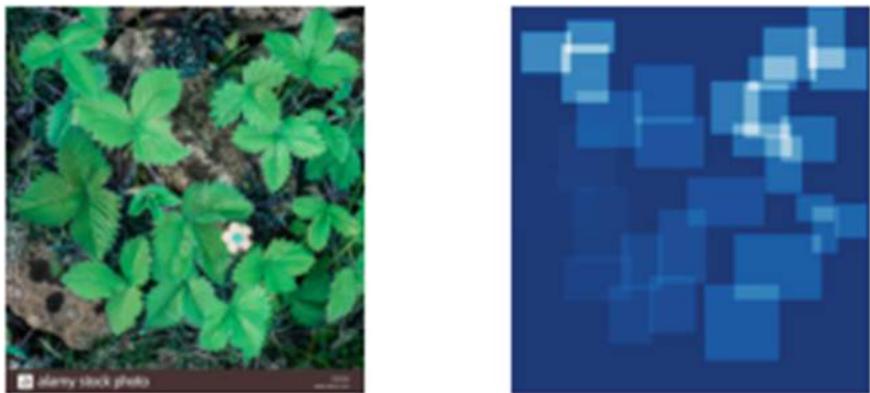


Fig. 1. Leaves bounding boxes

2.5 Disease Classification

The original PlantVillage dataset was selected instead of other third-party datasets to allow for more freedom in the customization and fine-tuning in the data augmentation step. Additionally, the dataset has leaf-segmented images, which is one of the main reasons behind the use of Mask R-CNN for leaf segmentation. Figure 2 below shows one image of each class present in the dataset:



Fig. 2. Leaf classes in the PlantVillage dataset [14]

The distribution of images between classes of the PlantVillage dataset used is unbalanced [15]. To tackle this issue, we will be using a function in Keras called class weights.

After calculating a weight for each class, the function will be able to backpropagate the loss according to the inverse of their distribution.

Before training, the dataset was split into the following sets: training, validation, and testing with each set having 80%, 10%, and 10%, respectively. Image augmentation will still be applied to increase the generalization of our model since lighting conditions can be very different from the labeled images and the ones captured from a field.

Some parameters such as shear, zoom, flip, and brightness have been chosen adequately. A sample of these augmentations is illustrated in Fig. 3:

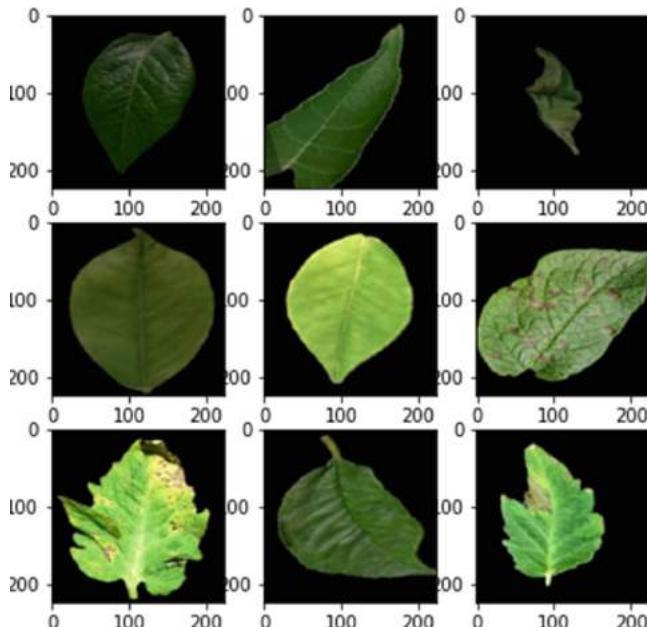


Fig. 3. Sample of augmented leaves for disease classification

The development of this work is performed according to the different stages shown in Fig. 4.

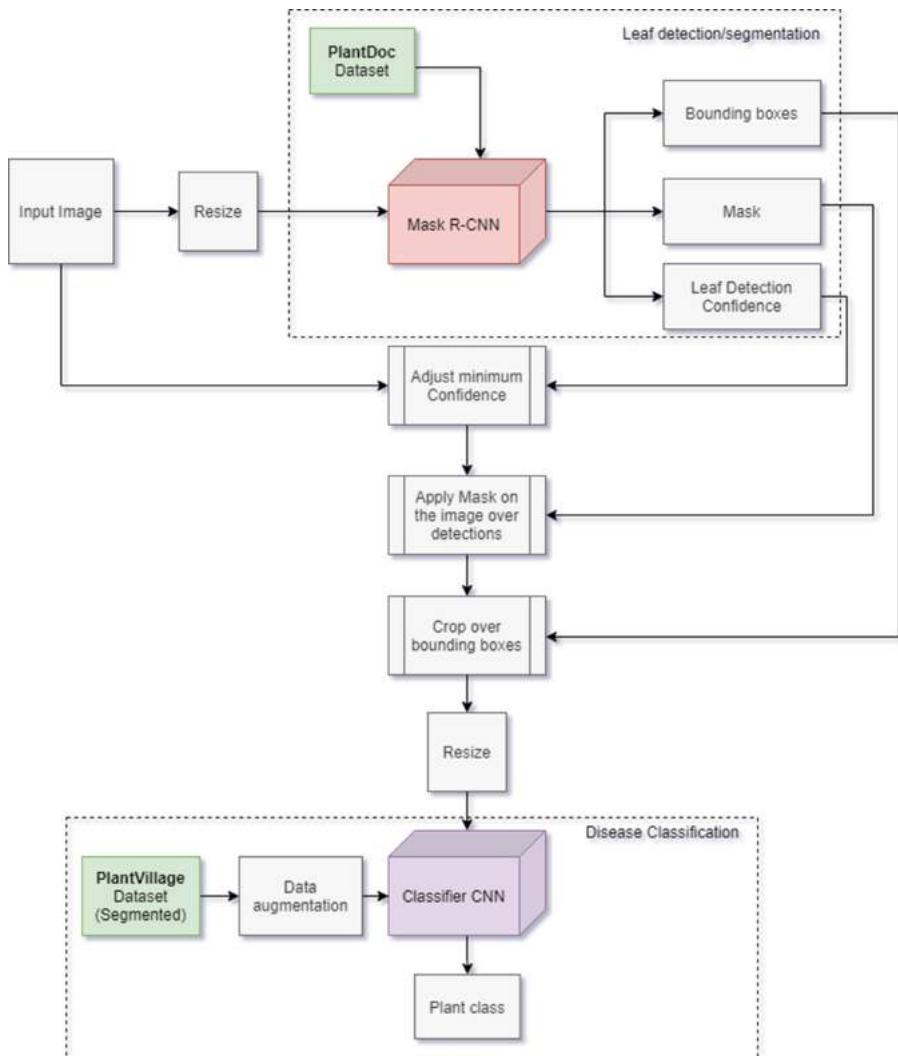


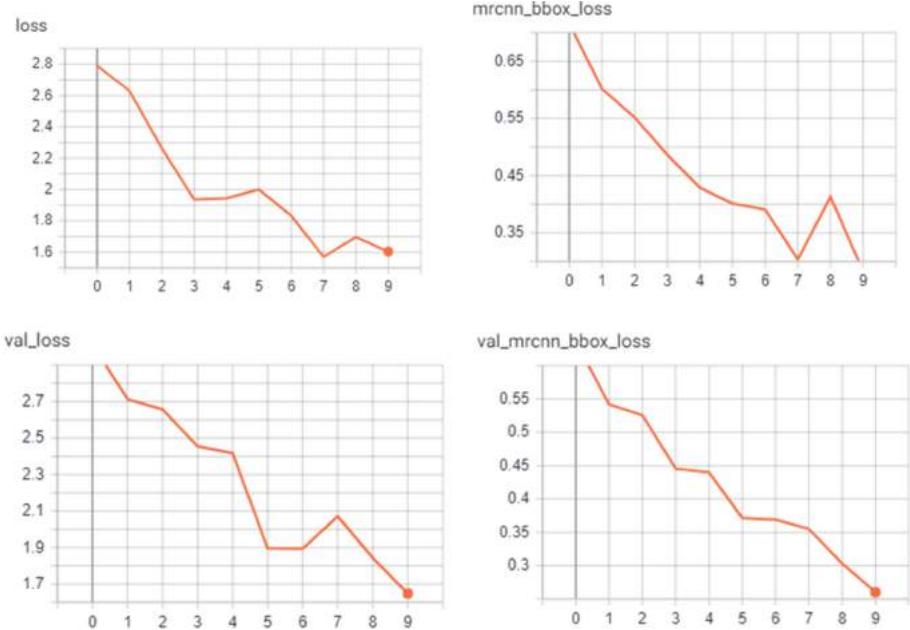
Fig. 4. Processing steps of the presented method

3 Results and Discussion

3.1 Leaf Detection

Mask R-CNN is trained with an input of size 256×256 , a learning rate of 0.001, a momentum of 0.9, minimum detection confidence of 0.8, and a mini mask of shape 56×56 . The model is trained for 10 epochs that lasted 2 h using the Google Colaboratory GPU and finished with a testing accuracy of 0.76%.

Figure 5 shows both the validation and training loss and bounding boxes loss of the Mask R-CNN model.

**Fig. 5.** Mask R-CNN losses

The images on the left of Fig. 6 represent the true bounding boxes while the images on the right show the predicted bounding boxes and leaves. We can see that the trained model was even able to detect leaves in the images that weren't present in the true bounding boxes which is a good sign that the model didn't overfit. For only 2 h of training and as a proof of concept, the results are better than anticipated. However, in most cases, the mask is not precisely covering the leaf. But this may be solved by fine-tuning other parameters such as the mask shape and disabling the mini mask (It was used to reduce training time).

3.2 Disease Classification

To speed-up training, a simple CNN was used with the following architecture: five convolutional layers with a kernel size of 3×3 and Relu activations shown in (1) combined with max-pooling every 2 layers and a SoftMax output layer which is defined by Eq. (2) [17]:

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \rightarrow 0 \end{cases} \quad (1)$$

where x is the input of a neuron

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \text{ for } i = 1, \dots, K \text{ and } Z = (z_1, \dots, z_K) \in \mathbb{R} \quad (2)$$

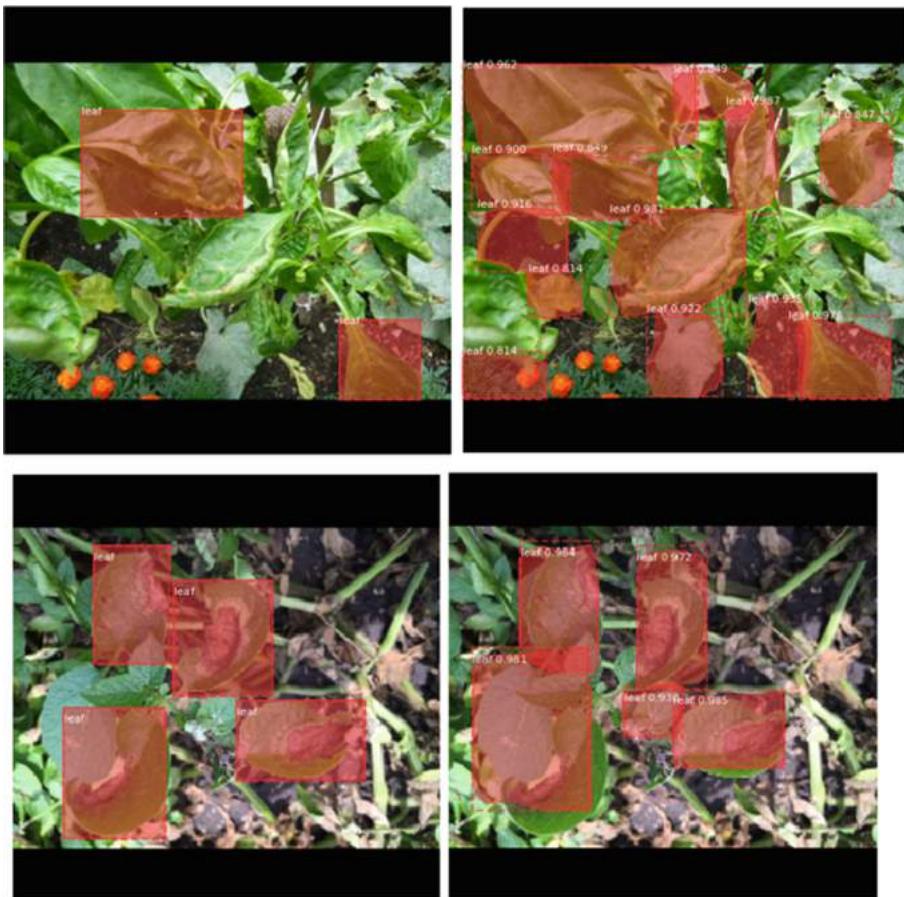


Fig. 6. True and predicted bounding boxes

where z_i is an element of the vector Z .

The loss is calculated using categorical cross-entropy with Adam optimizer. Figure 7 shows both validation and training accuracies. The validation accuracy significantly decreased over the last few steps confirming the overfitting of the model. To deal with such an issue, early stopping was used allowing us to continuously save the weights of the best performing model and return to that model at the end of the training.

The confusion matrix shown in Fig. 8 gives details on the performance of the model in differentiating between the classes. The y axis represents the true labels and the x axis represents the predicted labels.

The weighted average testing accuracy across all the classes is 83%. From the confusion matrix above, we note that most of the predictions of the model are correct apart from few exceptions such as a confusion between healthy soybean, healthy potato (91%), healthy raspberry (31%), and healthy cherry (34%). The misclassifications are mainly

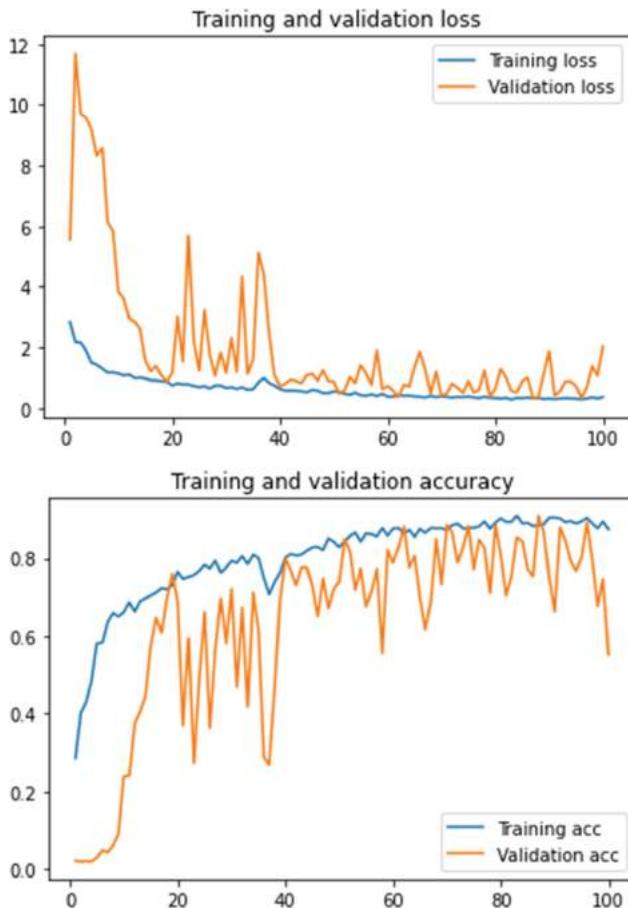


Fig. 7. Loss and accuracy for both validation and training steps

due to the close texture and shape between the leaves. The model would probably perform better if other architectures and fine-tuning methods are explored.

In another experiment, a model with a much lower complexity was trained. The images used were of size 50×50 and the architecture consisted of only 2 convolutional layers connected directly to the softmax layer. Surprisingly, the results were very comparable to the previous model. With a weighted average accuracy of 81% and a training time of only 8 min, it is clear that the disease classification can be achieved with simple and fast models.

To see the applicability of this method, few random plant images were gathered from the internet (Not existing in both datasets) were tested, and gave fairly good results. But it was found that the reason behind the wrong predictions is the poor segmentation.

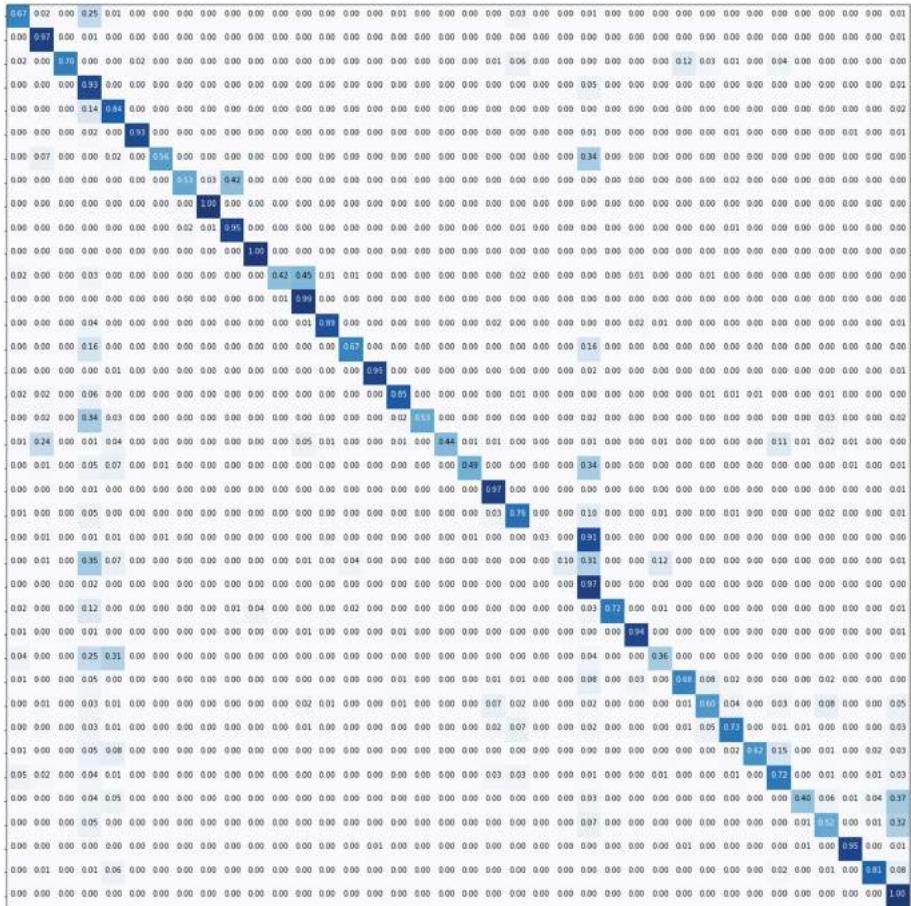


Fig. 8. Confusion matrix of the disease classifier

We can confirm that having separate models for each task is more time efficient and scalable. It is much faster to train the classification model alone than training Mask-RCNN for both tasks. Also, this method gives us more room for upgradability, because the segmentation model doesn't need to be retrained every time a new category is added.

The segmentation process could increase the performance of classification. However, this couldn't be proven in this work since the segmentation task of Mask R-CNN didn't perform very well. This method also suffers from a few limitations. One of them is that the classification fails when leaves contain multiple diseases or when they are too small/large. Also, the method relies on the availability and quality of the segmented images for classification.

4 Conclusion

In this paper, an approach for plant leaf segmentation and disease classification was developed based on computer vision and convolutional neural networks. Two datasets are used to demonstrate the ability of the proposed approach to detect and classify plant diseases.

The discussed approach can be used in agricultural fields to help farmers improve their crop quality and yield. By embedding the model in a mobile application or with the help of a UAV, the farmer can have cheap and fast monitoring of his crops.

In future work, a new segmentation method should be explored. In parallel, the results of Mask R-CNN can be improved by exploring other configurations, and training on a dataset that combines a large variety of sources such as weather conditions and locations.

References

1. Mohanty, S., Hughes, D., Salathé, M.: Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **7** (2016). <https://doi.org/10.3389/fpls.2016.01419>
2. UNEP. Smallholders, Food Security, and the Environment. Rome : International Fund for Agricultural Development (IFAD) (2013). <https://www.ifad.org/documents/10180/666cac2414b643c2876d9c2d1f01d5dd>
3. Abdu, A., Mokji, M., Sheikh, U.: Machine Learning for Plant Disease Detection: Investigative Comparison Between Support Vector Machine and Deep Learning (2020). <https://doi.org/10.11591/ijai.v9.i3>
4. Fang, Y., Ramasamy, R.: Current and prospective methods for plant disease detection. *Biosensors* **5**(3), 537–561 (2015). <https://doi.org/10.3390/bios5030537>
5. Reddy, P.R., Divya, S.N., Vijayalakshmi, R.: Plant disease detection technique tool—a theoretical approach. *Int. J. Innov. Technol. Res.* **4**, 91–93 (2015)
6. Chaudhary, P., Chaudhari, A.K., Cheeran, A.N., Godara, S.: Color transform based approach for disease spot detection on plant leaf. *Int. J. T. N*
7. Tete, T.N., Kamlu, S.: Detection of plant disease using threshold, k-mean cluster and ann algorithm. In: 2017 2nd International Conference for Convergence in Technology (I2CT), Mumbai, pp. 523–526 (2017). <https://doi.org/10.1109/I2CT.2017.8226184>.
8. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep neural networks based recognition of plant diseases by leaf image classification. *Comput. Intell. Neurosci.* **2016**, 1–11 (2016). <https://doi.org/10.1155/2016/3289801>
9. Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, N.: PlantDoc, Proceedings of the 7th ACM IKDD CoDS and 25th COMAD (2020). <https://doi.org/10.1145/3371158.3371196>
10. How MaskRCNN Works? | ArcGIS for Developers. Developers.arcgis.com. <https://developers.arcgis.com/python/guide/how-maskrcnn-works/>. Accessed 8 Aug 2020
11. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV) (2017). <https://doi.org/10.1109/iccv.2017>
12. Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. *Insights Imaging* **9**(4), 611–629 (2018). <https://doi.org/10.1007/s13244-018-0639-9>
13. Pawara, P., Okafor, E., Schomaker, L., Wiering, M.: Data Augmentation for Plant Classification. *Adv. Concepts Intell. Vis. Syst.* pp. 615–626 (2017). https://doi.org/10.1007/978-3-319-70353-4_52

14. PlantVillage Disease Classification Challenge, crowdAI (2016). <https://www.crowdai.org/challenges/1>. Accessed 04 Sep 2020
15. Mohanty, S.: spMohanty/PlantVillage-Dataset. GitHub (2016). <https://github.com/spMohanty/PlantVillage-Dataset>. Accessed 4 Sept 2020
16. Kuznichov, D., Zvirin, A., Honen, Y., Kimmel, R.: Data Augmentation for Leaf Segmentation and Counting Tasks in Rosette Plants. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2019). <https://doi.org/10.1109/cvprw.2019.00314>
17. Goodfellow, I., Bengio, Y., Courville, A.: 6.2.2.3 Softmax Units for Multinoulli Output Distributions. Deep Learning. MIT Press, pp. 180–184 (2016). ISBN 978–0–26203561–3



Medium Resolution Satellite Image Classification System for Land Cover Mapping in Nigeria: A Multi-phase Deep Learning Approach

Nzurumike L. Obianuju¹(✉), Nwojo Agwu¹, and Onyenwe Ikechukwu²

¹ Nile University of Nigeria, Abuja, Nigeria

² Nnamdi Azikiwe University Awka, Awka, Nigeria

Abstract. Deep learning-Convolutional Neural Network (CNN) algorithms have made considerable improvements beyond the state-of-the-art records for automatic classification of satellite images for land cover mapping. However, its use brings forward significant new challenges for developing countries such as Nigeria. This is majorly due to the scarcity of labeled high resolution dataset, as training a high performing model requires images with fine details as well as huge amount of labeled data to learn from. In this paper we designed and implemented a novel deep learning framework that can effectively classify pixels in a medium resolution satellite image into 5 major land cover classes. To achieve this purpose, we developed a CNN model from a pretrained ResNet-50, retrained using EuroSAT, a large-scale medium resolution satellite dataset. We further integrated Augmentation and Ensemble learning techniques to the model to create a unique model able to generalize to unseen data. To test the performance of the model, we created image patches from satellite images of cities in Nigeria (NigSAT). The final classification results shows that our model achieves a 96% and 80% accuracy on EuroSAT and NigSAT test data respectively. The model is expected to improve the efficiency and accuracy of satellite image classification tasks in developing countries where there is scarcity of large-scale training dataset, thus providing a more reliable and faster access to land cover information.

Keywords: Satellite image classification · Medium resolution · Deep learning

1 Introduction

Land cover mapping details how much of a region is covered by natural or artificial features such as forests, water bodies, bare lands, built-up, farmland, etc. The size of these features is constantly changing and are driven by a combination of both natural and anthropogenic causes. These changes can have significant impacts on people, the economy, and the environment. The impact is made worse in most developing countries due to population explosion, poverty, high rural-urban migration, etc. Furthermore, Land cover information is an important variable for many studies involving the Earth surface

and supports a wide range of applications including land use planning, forest monitoring, natural disaster monitoring, population estimation, geology, agricultural purposes. It is therefore very crucial for developing countries such as Nigeria to have information not only on existing land cover but also the capability to monitor the dynamics of land cover change in real time. This in turn will support relevant policies and decision-making for sustainable planning and development.

Satellite images of the earth taken from the air and from space shows a great deal about the planet's landforms, vegetation, and resources. Previous studies have shown that it offers the unique opportunity to investigate the physical features of the earth surface known as land cover mapping without the need to physically access the area. As at today, it is one of the most important tools for mapping the land-cover of the earth. However, the problem is in interpreting and analyzing the content of satellite imagery otherwise known as satellite image classification. This task has posed serious challenge in the past due to the complex and heterogeneous nature of satellite images.

With the advent of Artificial intelligence (AI), a field in computer science which develops smart programs to solve problems in ways that appear to emulate human intelligence, new attention is being paid to its application to satellite image classification. As a broad subfield of AI, machine learning (ML) is concerned with algorithms and techniques that allow computers to analyze data, learn from the data and make prediction on new data using computational and statistical methods. Traditional ML techniques such as unsupervised, supervised and object-based image classification were proposed in the past [1]. These techniques were designed primarily on the basis of spectral features reflected by the physical properties of the material and requires prior feature extraction, however, with any classification problem detecting good features/attributes can be difficult [2]. Other notable disadvantages of these methods include low accuracy, huge training time, difficulty in understanding the structure of the algorithms and inability to generalize well for a large-scale learning problem.

Recently, Deep learning (DL) models have emerged as a powerful solution to approach many ML tasks including satellite image classification [3]. It has become one of the most widely accepted emerging technology and has offered a compelling alternative to traditional classification techniques. It has the ability to handle the growing earth observation data. Studies have shown that among the many deep learning-based methods convolutional neural network (CNN) is the most established algorithm commonly used for classifying visual images. Its potentials in satellite image classification have been ascertained by many researchers in recent times for solving problems in the domain of geological mapping, land cover mapping, urban planning, geological image classification, population estimation etc. The strength of CNN models lies in three important factors: (i) large-scale labelled training data, (ii) high resolution satellite data (iii) and a good graphical processing unit (GPU).

However, labelled training data is very scarce, making it difficult for Convolutional Neural Networks (CNN) to be used for satellite image classification tasks [4]. To mitigate the problem of limited labelled training data, most researchers have used several techniques including transfer learning to leverage on either satellite image data repositories such as UC Merced (UCM) [5], EuroSAT [6] and more recently BigEarthNet

[7] or using natural image data repositories such as ImageNet or CIFAR-10. Data Augmentation (DA) has also been used extensive to artificially expand the available training dataset [3]. However, these techniques can only be applied in developed countries where there are at least a reasonable number of training datasets.

While, freely accessible medium resolution satellite images can be downloaded from earth observation programs such as European space Agency's (ESA) and the United States National Aeronautics and space Administration's (NASA), it is very expensive in terms of expertise, money and time for developing countries like Nigeria to label the right quantity of satellite image patches needed to train a high performing CNN model. This factor brings forward a significant new challenge for the application of deep learning in satellite image classification tasks in developing countries where there are no labelled training data repository. Additionally, most of the available models for satellite image classification were trained on high resolution datasets as a result cannot guarantee an accurate result when used for low- medium resolution satellite dataset due to the heterogeneous appearance of satellite images, variation in geography and difference in spatial details of the training datasets with the actual (target) dataset. There is need to develop robust CNN models that can be used in cases of non-availability of training data.

Inspired by the above, the main contributions of this paper are to:

1. Design and implement a novel deep convolutional neural network model that can accurately classify medium resolution Nigerian satellite (NigSAT) dataset for land cover mapping in Nigeria using the integration of three important techniques: Transfer learning, Augmentation and Ensemble learning. Our focus is on medium resolution data due to their massive use as the primary source of data for land cover change analysis research in Nigeria.
2. We created land cover image patches from satellite images of Nigeria (NigSAT) for testing and evaluating the CNN models.
3. Additionally, we detailed the methodology which can be followed for creation of more NigSAT image patches.

The remainder of this article is arranged as follows: Sect. 2 presents review of related work; Sect. 3 discusses the methodology of the proposed work. Section 4 presents the experimental results and analyses and Finally Sect. 5 provides the conclusion.

2 Review of Related Works

CNN, a deep learning model has extensively been used in computer vision applications in recent years and have achieved state of the art results in image classification tasks [8]. They are based on an end-to-end learning process, from raw data such as image pixels to semantic labels. Recent studies have demonstrated that advancement in sensor technology, computation power and deep learning-convolutional neural network model provides a powerful combination for automatic land cover mapping. However, a number of notable challenges are associated with its use in satellite image classification in developing countries. In this section we reviewed recent and relevant papers to highlight

the different advances made in designing CNN models with improved generalization performance on satellite image classification tasks. Our focus is on cases with limited or non-availability of training data. The findings of this preliminary study are discussed below under three headings:

2.1 CNN Model

The three main types of layers for building a CNN model are Convolutional Layer, Pooling Layer and Fully-Connected Layer. The arrangement of these layers plays a fundamental role in designing new models and also boosts its ability to learn complex representations. Different architectural designs have been proposed in the past and successfully used ranging from AlexNet, GoggleNet, InceptionV3, VGG-Net, ResNet etc. Each tried to overcome the shortcomings of previously proposed architectures in combination with new structural reformulations [9]. They majorly focused on increasing the depth of the layers which is aimed at improving the network structure [10]. Other popular techniques such as dropout, regularization and batch normalization have also been used to increase optimization as well as the overall performance of the models.

Most of these earlier state-of-the-art models were used to classify large scale everyday natural images [11] which significantly differs from satellite images. These models are not solely suitable for handling satellite datasets due to its complexity, large spectral bands or its inherent high variability. Additionally, estimating millions of parameters of a deep CNN requires large number of labelled samples, thereby preventing such models to be applied directly to satellite image tasks with limited training data.

Some researchers have proposed novel models: W. Zhang et al. [12] proposed Capsule network (CapsNet) which makes use of group of neurons as a capsule or vector to replace the neuron in the traditional neural network. These neurons can encode the properties and spatial information of features in an image to achieve equivariance, thus solving the problem of overfitting common to deep learning models. It also makes use of dynamic routing instead of max polling as seen in a regular CNN. Paoletti et al. [13] proposed a deep CNN based on 3D convolution, M. Carranza-García et al. [3] proposed a 2D CNN while H. Song et al. [10] proposed a patch-based light CNN (LCNN). To reduce the amount of parameters as well as the computational cost, they omitted the fully connected layer and the pooling layer in the architecture.

However, one consistent problem with these models is the absence of large-scale dataset for training the model. As a result, most of the models are shallow. Experimental results shows that deep models have an advantage over shallow models when dealing with complex data such as satellite image. Today, few researchers dare to train a model from scratch. Most recent research work are now focused on building on already existing models (pre-trained models) using transfer learning and implementing architectural innovations or using several other techniques to cope with the limited satellite datasets.

2.2 Transfer Learning

Despite CNNs' robust feature extraction capabilities, in practice it is difficult to train with small quantity of data. It needs to learn features from a large number of training data to achieve satisfactory model accuracy. An extensive review by Ball et al. [14] ranked

insufficient training dataset as a major challenge in the application of deep learning to the classification of remote sensing (satellite) imageries.

Transfer learning (TL) is one of the most popular Machine learning technique for solving problems with limited training data. It involves training a CNN model on a base dataset for a specific task then using the learned features/ model parameters as the initial weights in a new classification task. These models with millions or even hundreds of millions of parameters are trained on huge volume of data. It relies on the fact that the features learned in the lower layers of a CNN, like edges, curves or color may be general enough to be useful for other classification tasks.

The use of TL has facilitated the application of CNN to some scientific fields that have less available data [15]. Very recent works have demonstrated that features learnt from successful pre-trained deep CNN models can also be transferred to satellite image classification tasks. For instance R. P. De Lima & K. Marfurt [15] investigated the performance of TL from CNNs pre-trained on natural images for remote-sensing scene classification versus CNNs trained from scratch only on the remote sensing dataset themselves. The result shows that the former outperformed the later. Also Masoud Mahdianpari et al., [16] presented a detailed investigation of the capacity of seven well-known state-of-the-art pre-trained, deep learning models for the classification of complex wetland mapping in Canada. They both concluded that learning can be transferred from one domain to another and therefore will provide a powerful tool for satellite image classification.

Zhong Chen et al. [17] utilized TL for Airplane detection in remote sensing images. They used VGG16, a pre-trained network as the base network and replaced the fully-connected layer with a secondary network structure and finally fine-tuned with the limited number of airplane available training samples. M. Xie et al. [4] used a sequence of transfer learning steps to design a novel ML approach using satellite image. The first transfer learning problem P1 is object recognition on ImageNet; the second problem P2 is predicting nighttime light intensity from daytime satellite imagery; the third problem P3 is predicting poverty from daytime satellite imagery.

Although using pretrained CNN and adapting it to a new target task is a valid alternative, a good strategy needs to be adopted for the success of the transfer process. There are different strategies to perform transfer learning based on the domain, task at hand, and the availability of data, this includes full training, feature extraction and fine-tuning. Using CNN as a feature extractor involves using extracted features from a pre-existing model to train a new model of our choice while in fine tuning, the weights of the pre-trained CNN model are preserved on some of layers and tuned in the others.

The traditional strategy of transferring pre-trained deep CNNs for remote scene classification is to freeze the shallow layers and fine-tune the last deep layers [18]. However, the question that arises is how does depth affect the features of satellite image in the transferring process? In a systematic review of transfer learning application for scene classification by R. P. De Lima and K. Marfurt [15]. They evaluated how dependent the transfer learning process is on the transition from general to specific features by extracting features in three different positions denominated as “shallow”, “intermediate”, and “deep” for each one of the pretrained models. The shallow experiment uses the initial blocks of the base models to extract features and adds the top model. The intermediate experiment extracts the block somewhere in the middle of the base model. Finally, the

deep experiment uses all the blocks of the original base model, except the original final classification layer. The experimental result proved that deep adaptation involving several layers and carefully fine-tuning obtained better results than shallow learning or training a CNN model with randomly initialized weights. the depth of transferred CNNs enhances the generalization power of CNN and guarantees a general hypothesis for remote scene classification. A careful fine-tuning which involves several layers of the model, provides very good result [18].

2.3 Data Augmentation

Data Augmentation is also a well-known technique that is used in machine learning to artificially expand the size of a training dataset by creating modified versions of images in the dataset [3]. It creates variations of the images using a range of operations such as geometric transformations e.g., rotation, image rescaling, flipping, translations, cropping, zooming and addition of noise. While photometric transformations changes the color channels with the objective of making the CNN invariant to change in lighting and color [19].

The augmented data will represent a more comprehensive set of possible features which will help to improve the ability of the model to generalize what they have learned to new images. The volume and diversity of training data are essentially important in training a robust deep learning model. However, results by M. Abdelhack [11] shows that while some image augmentation techniques commonly used in natural image training can readily be transferred to satellite images, some others could actually lead to a decrease in performance. It is useful to consider the nature of satellite data when considering on an augmentation strategy.

D. I. N. Ore [20] and Stivaktakis. R et al. [21] proposed a Data Augmentation Enhanced CNN Framework to enhance the volume and diversity of remote sensing datasets by introducing three operations: flip, translation and rotation. The operations were used because they do not increase the spectral or topological information of the satellite images which is important for a consistent classification result. Krizhevsky et al. [22] proposed a data augmentation method to alter intensities of the RGB channels of raw data and achieved improved performance on the ImageNet benchmark.

A more recent data augmentation method for training CNN's known as Random Erasing was introduced by Zhong et al. [16]. The technique is specifically aimed at proffering solution to the problem of occlusion which occurs when part of an image is blocked off. These blocked images when used for training causes a model to generalize poorly when tested with unseen data (overfitting). Random erasing works by masking different rectangular portions of an image with random values thus generating training images with various levels of occlusion. This technique reduces the risk of overfitting and makes the model robust to occlusion. J. M. Haut et al. [23] adopted this data augmentation technique to mitigate the problem of overfitting in hyperspectral satellite image (HSI) classification.

Generative Adversarial Networks (GANs) a class of Neural Network also offers exciting opportunity to augment training dataset and help boost accuracy in image classification. GAN offers a way to unlock additional information from a dataset by generating synthetic samples with the appearance of real images. Various variations of GAN

has been proposed and experimental results proved its effectiveness [24]. Its limitation however is that GANs cannot be relied upon to produce images with perfect fidelity as seen in traditional augmentation techniques.

Some researchers used a combination of TL and augmentation techniques to design novel architectures. Hans et al. [25] proposed a two-phase method combining TL and web data augmentation to classify natural images. With their method, the useful feature presentation of pre-trained network was efficiently transferred to a target task, and the original dataset was augmented with the most valuable internet images. A work similar to ours was done by Grant. J. Scott et al. [26]. They investigated the use of Deep CNN for land-cover classification in high-resolution satellite imagery. To overcome the lack of massive labeled data sets, they utilized two techniques in conjunction with DCNN: TL with fine-tuning and data augmentation tailored specifically for remote sensing imagery.

In summary, the techniques outlined above have been seen to work in cases where there are at least few training datasets to be expanded, used to fine-tune a pre-trained model or both. In contrast to existing satellite image classification research works which focuses on finding solutions for improving CNN algorithms in case of limited training data, we propose a multi-phase Deep learning approach method which works in cases of non-availability of training data. It also provides additional functionalities by integrating Ensemble learning techniques to transfer learning and augmentation techniques. A system that can extract useful information from medium resolution satellite images is highly desirable especially in developing countries.

3 Methodology

The goal of this research is to propose a model for classifying pixels in medium resolution satellite images of Nigeria into five land cover classes for land cover mapping, using a multi-phase learning approach. First, we fine-tuned and retrained a pretrained ResNet-50 using EuroSAT, a large-scale medium resolution dataset. We further integrated Augmentation and Ensemble learning techniques to the model to create a unique model able to generalize to unseen NigSAT data. To test the performance of the model, we created image patches from satellite images of cites in Nigeria (NigSAT). We describe the data and the preprocessing steps that we used in Sect. 3.1, the CNN model architecture choices in Sect. 3.2, and the training methodology that we follow to train, validate, and test our models in Sect. 3.3.

3.1 Dataset

Two sets of datasets were used in our study.

3.1.1 EuroSAT Dataset

EuroSAT a benchmark dataset for land cover classification was used for training, testing and validating our model. Introduced by Patrick Helber et al. [6], EuroSAT are Sentinel-2 satellite labeled and geo-referenced image patches measuring 64×64 pixel gotten from 34 countries in Europe. This dataset originally consists of 10 land use and land cover

(LULC) classes with each class between 2000 and 3000 and a total of 27,000 divided into 10 generic land cover types. We reclassified similar land cover classes from the EuroSAT dataset to suit our land cover classes of interest as shown in Table 1.

Table 1. Land cover class reclassification

land cover used in EuroSAT	New re-classified land cover classes	Class label	Total number after reclassification
Highway	Bare land	1	2500
Residential and industrial	Built-up	2	5500
Annual crop, herbaceous vegetation, pasture, permanent crop	Farm land	3	10500
Forest	Forest	4	3000
Rivers, lakes, sea	Water bodies	5	5500

To prevent class imbalance among the different land cover classes, the lesser classes were augmented as shown in Table 2.

Table 2. Dataset statistics for training, validating and testing after data augmentation

Classes	Augmentation		(Split 80%, 10%, 10%)		
	Before	After	Training	Validation	Testing
Bare land	2500	10502	8401	1050	1051
Built-up	5500	10483	8385	1048	1050
Farmland	10500	10501	8401	1050	1050
Forest	3000	10501	8400	1050	1051
Water bodies	5500	10503	8401	1050	1052

3.1.2 NigSAT Dataset

To evaluate and test the ability of our model to classify NigSAT images correctly, we generated image patches from satellite image of Abuja, Nigeria. The methodology is described below.

- **Download:** We downloaded medium resolution Sentinel-2A satellite images of Abuja via their website <https://scihub.copernicus.eu/dhus/#/home>. The Sentinel-2 image was acquired in February 2020. The considered tile is distributed over the six area councils in Abuja-Nigeria (Abaji, Abuja Municipal, Bwari, Gwagwalada, Kuje and Kwali).

- **Image Processing/preparation:** Sentinel-2 captures multispectral imagery spanning 13 different bands with three different resolutions 60m, 20m and 10m. We decided to use the bands with 10m resolution (2, 3, 4 and 8). Furthermore, the 10m resolution imagery provides the greatest level of detail for medium resolution data. A color composite is created by combining the three raster image bands in ArcGIS 10.5 version as shown in Fig. 1.

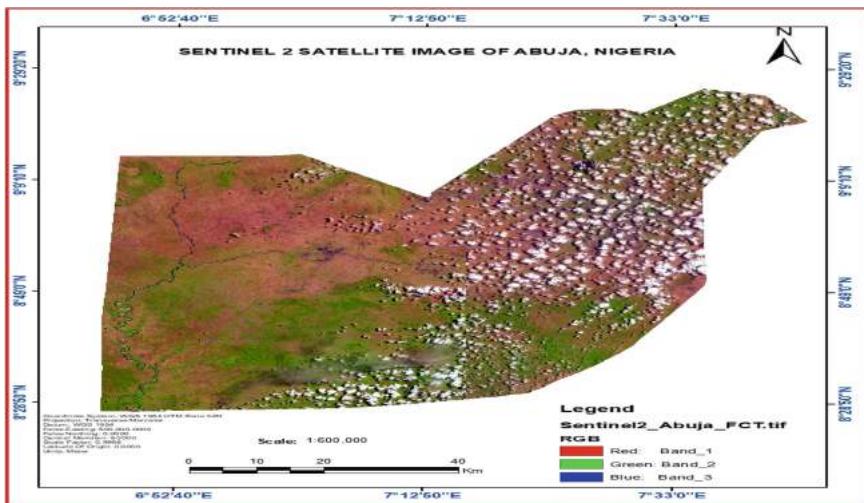


Fig. 1. Satellite image of Abuja, Nigeria

- **Image Enhancement:** The image was brightened to produce a clearer and relevant image detail.
- **Extracting and labeling patches:** This aims to group pixels with common spectral characteristics into a given category. We classify the images into the five land cover classes (Farm land, Built up, bare land, Water bodies and Forest). A total of 25 image patches were extracted, five for each class. These extracted features are in form of image patches of size 64×64 in Geo-TIFF format. We resized all the image patches to the dimension requirement for ResNet-50 model which is (224×224) . Samples from the two datasets are shown in Fig. 2.

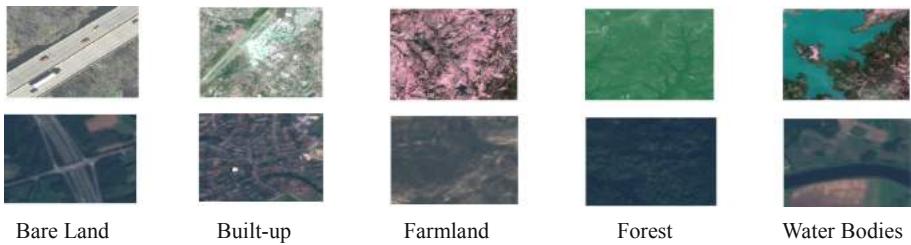


Fig. 2. EuroSAT and NigSAT image patches

3.2 The Model Framework

Our model framework is based on Transfer Learning, Augmentation and Ensemble Learning as shown in Fig. 3.

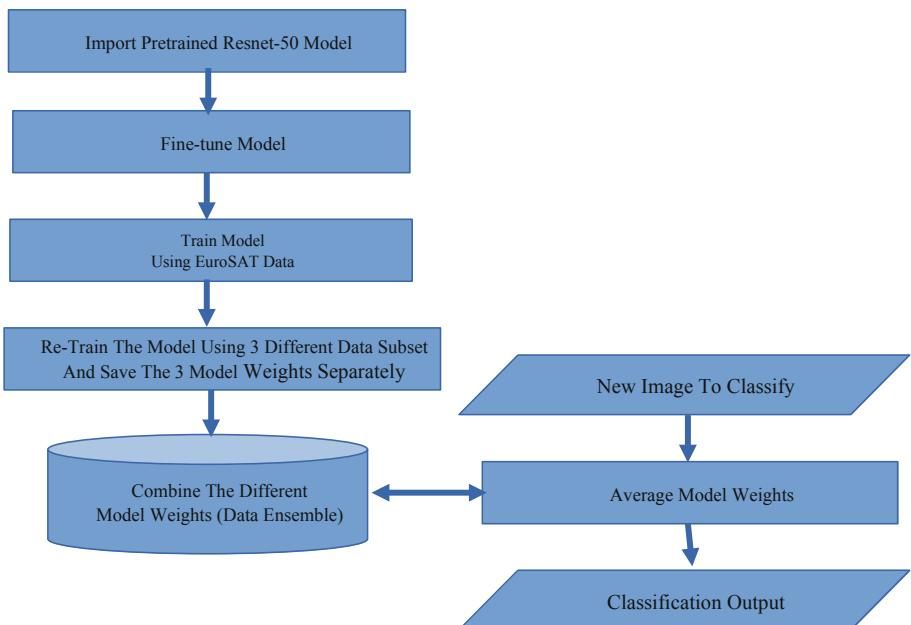


Fig. 3. Model framework

We first import a pretrained ResNet-50 model, a 50-layer variant designed by Microsoft Researchers in 2017. The most important characteristics when considering choice of a pretrained model are network accuracy, speed, and size. After the celebrated victory of AlexNet, ResNet short for Residual Network was the most groundbreaking model in the computer vision/deep learning community. See [27] for more details on the structure and workings of a ResNet model. Also considering our data, choosing a deeper network was crucial to extract more features.

Our model architecture consists of the base ResNet50 model, with the top layers replaced with a two densely connected layers (of size 1024 and 5, respectively), separated by a size 0.5 (50%) dropout layer which zeros out the activation values of randomly chosen neurons during training and makes our model to learn more robust features. the model architecture is represented as shown in Fig. 4.

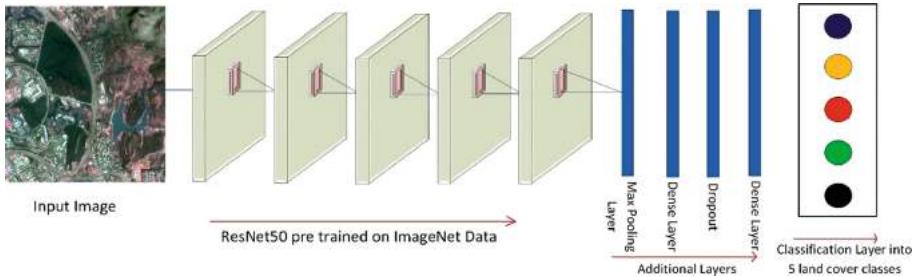


Fig. 4. Model architecture

3.3 Training the CNN Model

The training phase is where the network "learns" from the data it will be fed with. First, we trained the ResNet50 architecture pre-loaded with ImageNet weights, on the reclassified EuroSAT dataset over 10 epochs. The training was done using RMSprop, a gradient based optimizer, at a learning rate of $1e-4$.

To further improve the performance and robustness of the trained model above to classify NigSAT dataset, we used Boosting, a technique in Ensemble learning. Ensemble learning allows us to train multiple models instead of a single model and then combine the models to make better prediction. We re-trained the model weight saved above with three different subsets of datasets **train_datagen1**, **train_datagen2** and **train_datagen3** to generate three separate model weights. The three subsets of datasets were generated from the original dataset using different data augmentation transformations derived from the Keras and TensorFlow libraries. The first augmentation focused on randomly applying traditional physical transformations, such as: Rescaling, rotations, flipping, zoom, etc. the 2nd utilized photometric transformations such as color space: Hue, saturation, brightness and contrast randomization. While the last concentrated on random erasing which aims at solving the problem of occlusion. It masks different portions of the image with rectangular region of arbitrary size thus, generating training images with various levels of occlusion. Finally, the weights generated from the different trained models were saved and combined. This was used to test the NigSAT dataset (Fig. 5 and 6).

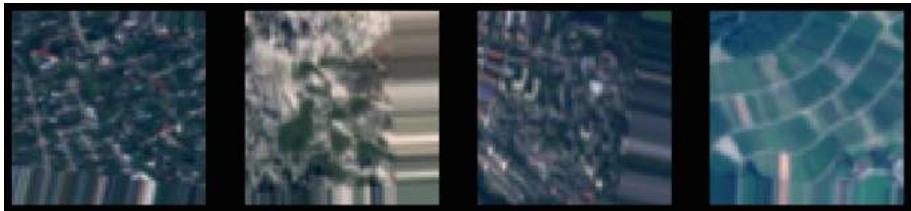


Fig. 5. Examples of images generated from train_datagen1.

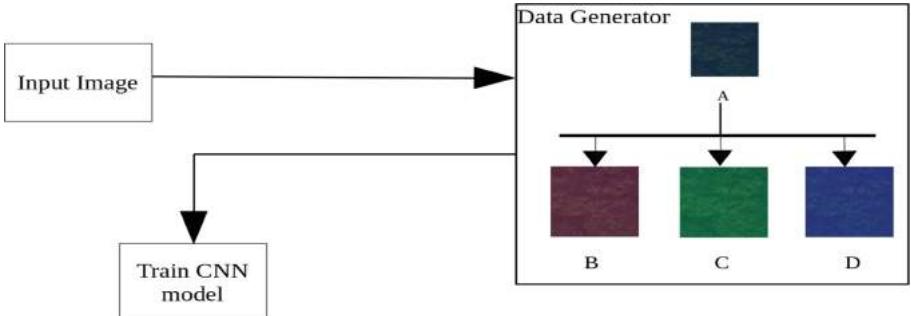


Fig. 6. Data augmentation - Different Color Transformation of Forest in train_Datagen2. (Color figure online)

4 Results and Analysis

This section analyzes the results obtained from each experiment in our study. We performed four experiments as follows:

4.1 To evaluate the effectiveness of the pre-trained ResNet50 architecture pre-loaded with ImageNet weights on the reclassified EuroSAT dataset (in-domain testing).

The training loss and accuracy are plotted. The model was used to predict the Validation and Testing datasets. We observed a performance accuracy of about 95% and 96% on the validation and testing dataset respectively. The results of the predictions on the test dataset are presented in the confusion matrix shown in Tables 3 and 4. Other performance measures used are precision, recall and F1 score.

Table 3. Confusion matrix for the EuroSAT test set

	Bare land	Built-up	Farmland	Forest	Water Bodies
Bare land	991	24	15	5	16
Built-up	4	1038	4	2	1
Farmland	7	1	1032	1	9
Forest	3	1	7	1014	26
Water bodies	22	2	11	11	977

Table 4. Classification result for EuroSAT test set

	Precision	Recall	F1-score	Support
Bare land	0.96	0.94	0.95	1051
Built-up	0.97	0.99	0.98	1049
Farmland	0.97	0.98	0.97	1050
Forest	0.95	0.96	0.96	1051
Water bodies	0.95	0.93	0.94	1052
Accuracy			0.96	5253
Macro avg.	0.96	0.96	0.96	5253
Weighted avg.	0.96	0.96	0.96	5253

4.2 Testing the Ability of the Model in 1 to Classify NigSAT Images (Out of Domain Testing)

The model was also used to predict NigSAT data, as expected we observed a poor accuracy of 48%. It shows a drop from 94% to 48%. We believe that this poor accuracy was caused by the Variation between the EuroSAT and NigSAT dataset (Tables 5 and 6).

Table 5. Confusion matrix for the NigSAT test set

	Bare Land	Built Up	Farmland	Forest	Water Bodies
Bare land	0	2	3	0	0
Built up	0	4	1	0	0
Farmland	0	2	2	1	0
Forest	0	0	0	5	0
Water bodies	0	2	1	1	1

Table 6. Classification result for the NigSAT test set

	Precision	Recall	F1-score	Support
Bare land	0.00	0.00	0.00	5
Built up	0.40	0.80	0.53	5
Farmland	0.29	0.40	0.33	5
Forest	0.71	1.00	0.83	5
Water bodies	1.00	0.20	0.33	5
Accuracy			0.48	25
Macro avg.	0.48	0.48	0.41	25
Weighted avg.	0.48	0.48	0.41	25

- 4.3 **With our problem defined**, we sort for ways to improve the generalization ability of the model so as to have a better prediction accuracy. We re-trained the model above with three different subsets of datasets train_datagen1, train_datagen2 and train_datagen3 to generate three separate model weights. A careful look at the recall and precision performance scores achieved by the weights shows that based on the Augmentation strategy used, each model was able to predict a particular land cover class better than the rest
- 4.4 Finally, using Ensemble learning technique, we combined the weights learned by the models. It was then used to test the NigSAT dataset. The results of the predictions are presented in the confusion matrix and classification reports shown in Tables 7 and 8, respectively. We observed a performance accuracy of about 80% gaining an accuracy of 32%.

Table 7. Confusion matrix for the NigSAT test set

	Bare land	Built-up	Farmland	Forest	Water bodies
Bare land	4	0	0	0	1
Built-up	0	4	0	0	1
Farmland	0	1	4	0	0
Forest	0	0	0	5	0
Water bodies	1	0	1	0	3

We implemented the model in Python using the Keras and TensorFlow deep learning libraries which provides a complete deep learning toolkit for training and testing models. In addition, it should be noted that all the experiments were performed on HP Envy 17t (10th Gen Intel i7-1065G7, 16GB DDR4, 1TB HD + 256GB NVMe SSD, NVIDIA GeForce 4GB GDDR5).

Table 8. Classification result for the NigSAT test set

	precision	recall	f1-score	support
Bare land	0.80	0.80	0.80	5
Built-up	0.80	0.80	0.80	5
Farmland	0.80	0.80	0.80	5
Forest	1.00	1.00	1.00	5
Water bodies	0.60	0.60	0.60	5
Accuracy			0.80	25
Macro avg.	0.80	0.80	0.80	25
Weighted avg.	0.80	0.80	0.80	25

4.1 Discussions

Our research work was able to show the following:

- a. That deep networks trained for image recognition in one task (ImageNet) can be efficiently transferred and used for satellite image classification. This was demonstrated on the EuroSAT dataset, where above 96% mean accuracy was achieved using a ResNet-50 model pretrained on ImageNet.
- b. We proved the high variability which occurs among satellite images of different locations which was demonstrated when the model trained with EuroSAT data was tested on NigSAT data yields a drop from 96% to 46% in accuracy.
- c. Augmentation techniques have the ability to produce training data more robust to achieve a high generalization ability.
- d. Ensemble technique can be successfully applied to Deep CNN models for satellite image classification.

5 Conclusion

The lack of training data in developing countries is a major obstacle to harnessing the potentials of DL in satellite image classification. In this preliminary research work, we attempted to design and implement a novel technique based on deep convolutional neural network for medium resolution satellite image classification. The aim of our research is to improve the classification accuracy and speed of satellite image classification for land cover mapping in Nigeria. The model was based on the integration of three important techniques: transfer learning, augmentation and ensemble learning. Five different land cover classes (Forest, bare land, farm land, built-up and water bodies) were classified. We used EuroSAT [6] a benchmark dataset for land cover classification for training, testing and evaluating our model. Additionally, we detailed the method for creating a

benchmark dataset from satellite images of cities in Abuja for evaluating our model. Although the size of the NigSAT dataset is small, it is the very first set of indigenous datasets that can be used to test a model in Nigeria. Also, the method can be adopted to create more dataset. The final classification results show that our model achieves up to 80% accuracy. However, the major challenges in classifying these NigSAT medium resolution satellite images are non-availability of large-scale high-resolution training data and excessive amounts of background noise including partial images of trees or cars present in the NigSAT datasets which was absent in EuroSAT dataset. The study also illustrated the need for indigenous large scale medium resolution satellite image patches to be produced for further research work. In future, we aim to experiment by training a model solely using NigSAT datasets. Also, many aspects of the proposed method have not yet been fully addressed and explored, and further investigations are still needed.

References

1. Karim, S., Zhang, Y., Asif, M.R., Ali, S.: Comparative analysis of feature extraction methods in satellite imagery. *J. Appl. Remote Sens.* **11**(04), 1 (2017)
2. Zhang, C.: Deep Learning for Land Cover and Land Use Classification Ce Zhang BSc, MSc, MSc, no. November 2018
3. Carranza-García, M., García-Gutiérrez, J., Riquelme, J.C.: A framework for evaluating land use and land cover classification using convolutional neural networks. *Remote Sens.* **11**(3), 274 (2019)
4. Xie, M., Jean, N., Burke, M., Lobell, D., Ermon, S.: Transfer Learning from Deep Features for Remote Sensing and Poverty Mapping (2014)
5. Yang, Y., Newsam, S.: Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification, no. May 2014
6. Helber, P., Bischke, B., Dengel, A., Borth, D.: Introducing eurosat: a novel dataset and deep learning benchmark for land use and land cover classification. *Int. Geosci. Remote Sens. Symp.* **2018**, 204–207 (2018)
7. Song, J., Gao, S., Zhu, Y., Ma, C.: A survey of remote sensing image classification based on CNNs. *Big Earth Data* **3**(3), 232–254 (2019)
8. Neware, R., Khan, A.: Survey on Classification Techniques Used in Remote Sensing for Satellite Images. In: Proceedings 2nd International Conference on Electronics, Communication and Aerospace Technology ICECA 2018, no. March, pp. 1860–1863 (2018)
9. Khan, A., Sohail, A., Zahoor, U., Qureshi, A.S.: A Survey of the Recent Architectures of Deep Convolutional Neural Networks, pp. 1–62 (2019)
10. Song, H., Kim, Y., Kim, Y.: A Patch-Based Light Convolutional Neural Network for Land-Cover Mapping Using Landsat-8 Images, pp. 1–19 (2019)
11. Abdelhack, M.: An Open-source Tool for Hyperspectral Image Augmentation in Tensorflow, pp. 1–4 (2020)
12. Zhang, W., Tang, P., Zhao, L.: Remote sensing image scene classification using CNN-CapsNet. *Remote Sens.* **11**(5), 494 (2019)
13. Paoletti, M.E., Haut, J.M., Plaza, J., Plaza, A.: ISPRS journal of photogrammetry and remote sensing a new deep convolutional neural network for fast hyperspectral image classification. *ISPRS J. Photogramm. Remote Sens.* **145**, 120–147 (2018)
14. Ball, J.E., Anderson, D.T., Ball, J.E., Anderson, D.T., Chan, C.S.: Comprehensive survey of deep learning in remote sensing : theories , tools , and challenges for the community, **11**(4) (2020)

15. De Lima, R.P., Marfurt, K.: Convolutional Neural Network for Remote - Sensing Scene Classification: Transfer Learning Analysis (2020)
16. Land, C., Mapping, C., Multispectral, U., Imagery, R.S.: Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery (2018)
17. Chen, Z., Zhang, T., Ouyang, C.: End-to-end airplane detection using transfer learning in remote sensing images. *Remote Sens.* **10**(1), 1–15 (2018)
18. Access, O.: We are IntechOpen , the world ' s leading publisher of Open Access books Built by scientists , for scientists TOP 1 % Utilization of Deep Convolutional Neural Networks for Remote
19. Taylor, L., Nitschke, G.: Improving Deep Learning using Generic Data Augmentation. no. October (2017)
20. Ore, D.I.N.: Deep Learning in Remote Sensing Scene Classification : A Data Augmentation Enhanced CNN Framework (2018)
21. Stivaktakis, R., Tsagkatakis, G., Tsakalides, P.: Deep learning for multilabel land cover scene categorization using data augmentation. *IEEE Geosci. Remote Sens. Lett.* **16**(7), 1031–1035 (2019)
22. Krizhevsky, B.A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks (2012)
23. Haut, J.M., Paoletti, M.E., Plaza, J., Plaza, A., Li, J.: Hyperspectral image classification using random occlusion data augmentation. *IEEE Geosci. Remote Sens. Lett.* **16**(11), 1751–1755 (2019)
24. Bowles, C., et al.: GAN Augmentation : Augmenting Training Data using Generative Adversarial Networks
25. Han, D., Liu, Q., Fan, W.: A new image classification method using CNN transfer learning and web data augmentation. *Expert Syst. Appl.* **95**, 43–56 (2018)
26. Scott, G.J., Marcum, R.A., Davis, C.H., Nivin, T.W.: Fusion of deep convolutional neural networks for land cover classification of high-resolution imagery. *IEEE Geosci. Remote Sens. Lett.* **14**(9), 1638–1642 (2017)
27. Mikami, H., Suganuma, H., U-chupala, P.: ImageNet / ResNet-50 Training in 224 Seconds, no. Table 2.



Analysis of Prediction and Clustering of Electricity Consumption in the Province of Imbabura-Ecuador for the Planning of Energy Resources

Jhonatan F. Rosero-Garcia^(✉), Edilberto A. Llanes-Cedeño,
Ricardo P. Arciniega-Rocha, and Jesús López-Villada

Universidad Internacional SEK, Quito-Ecuador, Instituto Superior Tecnológico
17 de Julio, Urcuquí, Ecuador
jfrosoro.mee@uisek.edu.ec

Abstract. The electrical planning of a country is a growing need for its economic development. However, this analysis is complex to carry out due to the different consumption habits of the participants in this market. In this sense, in Imbabura-Ecuador, residential electricity consumption is the one that needs the greatest electricity demand. For this reason, a prediction and grouping analysis by municipalities is presented using machine learning algorithms in order to determine consumer trends and present reports for proper electrical planning. As relevant results, the models of decision support machines and random forests proved to be suitable for this task with a prediction error of less than 10%. For its part, the k-means algorithm was able to group four types of electricity consumption with a representation of 98% of the data variability.

Keywords: Electrical consumption · Electrical analysis · Regression models · Clusterins

1 Introduction

One of the main components of the economic development of a country is its capacity to produce electricity reliably and at the lowest possible cost. For this, it is necessary to determine the current and future behavior of electricity demand for proper planning that must consider political, economic, social, environmental and technological variables [14]. However, these analyzes must be undertaken by government entities and the private sector in order to implement programs and plans that have direct impacts on electricity demand. This must be closely related to the demographics of each country in conjunction with the different renewable or non-renewable natural resources to be used as sources of electricity production [2].

In this sense, the forecast of electricity consumption allows the cooperation of all participants in this market to cooperate effectively with each other. In this

way, the industrial sector can obtain warnings about its excessive consumption and prevent electricity supply companies from producing electricity optimally by avoiding overloading of generation equipment [18]. For its part, in the residential sector, the analysis of electricity demand is directly related to the expansion of urban sectors within a city. With this, planning on the need for the implementation of transformers and the installation of electric poles becomes a technical task. However, this electrical forecasting process is a complicated task to carry out due to the large amount of data obtained from users and the different variables that this implies [9]. This causes that the electrical system works with certain uncertainty that affects the cost of its production. Taking into consideration that the electrical energy produced on a large scale cannot be stored [7]. In addition, electricity consumption is variable due to its high demand at different times. Another point to consider is the scarce knowledge of electricity consumption trends in the different sectors of a country. This is of great importance to avoid an over-adjustment of policies and projects in the electricity sector. Since when establishing groups of electricity consumption, its planning does not focus on particularities and seeks a generality in future electricity generation plans. Under this criterion, the electricity demand curve allows representing the real demand vs. the expected one. This process must be carried out using complex adaptive algorithms that permanently collect information [19]. However, this analysis must be seen in a different way between the demographics of each country and the type of consumer.

With the aforementioned, Ecuador has a great electricity generation capacity by using the different natural resources it has. Since its electricity consumption has constant annual growth. In 2019, it was 4.5% [1], this results in the need to implement decision support systems that have the ability to forecast electricity consumption that best adjusts to the real values and, if applicable, recommend actions on changes in habits consumer. For this reason, there are large data repositories for analysis. This process can only be carried out under robust criteria for machine learning hosted on servers with high computational resources [20]. Therefore, it is proposed to carry out an exhaustive study of the available information, to find consumption trends in the different provinces of the country, to forecast consumption for each one of them [21]. With this, you can group them according to their main characteristics and establish relationships between them.

This work is focused on forecasting the electricity consumption by municipalities within the province of Imbabura-Ecuador in relation to the different customer characteristics consumption collected by the companies that supply electricity service. To do this, available information is collected from the Agency for the regulation and control of energies and non-renewable natural resources, subscribed to the Ministry of Electricity of Ecuador. With this, a data analysis scheme will be made with different criteria to determine the best solution to be implemented. For this, it begins with a data cleaning process, subsequently, a comparison of multivariate linear regression algorithms is carried out to determine the model that best adapts and has a better capacity to predict consumption. In addition, a grouping is made between the municipalities to establish

consumption habits. All this is seen in a graphical interface using a decision support tool for adequate decision support by the country's regulatory entities. As relevant results, the models of decision support machines and random forests proved to be adequate for this task with a prediction error of less than 10%. For its part, the k-means algorithm was able to group four types of electricity consumption with a representation of 98% of the variability of the data.

The rest of the document is structured as follows: Sect. 2 presents the works related to the issue raised. Materials and methods are shown in Sect. 3. Section 4 presents the results. Finally, the conclusions and future work are shown in Sect. 5.

2 Related Works

This topic has been analyzed in works such as [8], which presents an analysis of data on household electricity consumption to be stored on a server in the cloud. In this case, they do not present a solution to a specific sector and focus on residential consumption. [15], carries out an architectural study in buildings and their influence on electricity consumption based on user trends. The authors in [5, 17] make use of mathematical models and simulations in different applications of electrical power systems. For its part, [3] uses time series and clustering to predict residential electricity consumption. Recently, [12] has published a set of data on electricity consumption in Colombia and the different characteristics of consumers. In Ecuador, works such as [11] carry out studies on electricity demand in general terms and its relationship with the CO₂ emissions that its production entails. [16] presents a solution to predict electricity consumption in public institutions. On the other hand, [6] uses neural networks to predict electricity consumption in the different cantons of the province of Pichincha-Ecuador. Finally [10] carries out a study of grouping by consumer characteristics in the same province of the previous work.

The aforementioned works have presented adequate solutions on electrical prediction and grouping by consumers. However, there are some open problems on the issue raised, such as the combination of various machine learning criteria, the formal presentation of results in interactive interfaces, the relationship with national electricity generation systems, among others. For these reasons, the present work covers these criteria by presenting mathematical models for the extraction of intrinsic knowledge present in the data set obtained on the parameters of electricity consumption in the province of Imbabura-Ecuador. With this, the presentation of results in a decision support tool seeks to integrate academic sectors with public entities for the generation of state policies based on reliable information.

3 Materials and Methods

This chapter shows, on the one hand, the description of the available database and its main characteristics. On the other hand, the proposed data analysis scheme is shown together with the information grouping prediction criteria.

3.1 Database Description

In the Imbabura province, its main business line is the production of wooden articles and services for medium-sized industries [13]. Related to electricity production, it has 11 medium-production plants that generate an effective power of 115 MW per year with a total of 130,000 subscribers from the residential sector with an annual consumption of 150 GWh. For this reason, Ecuador has an interconnected electrical system to supply the missing energy to the provinces through its emblematic projects such as the hydroelectric plant *Coca Codo Sinclair* [1, 4]. In this sense, the information acquired through the open data repositories provided by Ecuador made it possible to acquire the consumption reports of residential customers from 2017 until 2020. This type of consumer is established since it is the largest in the province. With this, we have a multivariate data matrix of the form $\mathbf{Y} \in \mathbb{R}^{m \times n}$, where m is the number of instances, and n is the number of variables. Therefore, we have $m = 3500$ and $n = 8$ which are the characteristics of: canton, municipality, type of electrical equipment (220 V electronic devices), number of customers by geographic area, billed energy, incremental consumption, residential consumption and billing.

3.2 Data Analysis Proposed

This data schema focuses on three main components. The transformation, since it is necessary to clean the data and update the information because the Spanish language has accents that are sometimes found in the databases that generate a greater number of erroneous attributes. In addition, there are empty or null data that would generate errors in the prediction and data grouping models (Sect. 3.3). On the other hand, the different multivariate linear regression models must be chosen to determine the appropriate one, these are shown in Sect. 3.4. Finally, the clustering algorithm selected is *k-means*, due to its great effectiveness in this task. However, it is necessary to properly choose the correct value of clusters within the proposed database [10]. All this process is seen in Fig. 1.

3.3 Transformation

The $\mathbf{Y} \in \mathbb{R}^{m \times n}$ matrix has three main drawbacks. The first is based on having null data or empty cells. For this, we proceeded to eliminate this data to avoid inconveniences. Second, you have categorical variables with a tilde. This causes the same variable to have different values because it has a different *ASCII* encoding. For this reason, a search method was carried out to eliminate all these words and standardize all the variables. This process was carried out with a *ETL* tool. Finally, it is necessary to code these categorical variables to train a multivariate linear regression model. To do this, the *One-Hot-Encoding* method has been used from the *sklearn* library of the *python* programming environment. This method is helpful because when you encode the variables, each category value is converted into a new column and a value of 1 or 0 (notation for true or false) is assigned to the column. This avoids misinterpreting numeric values as

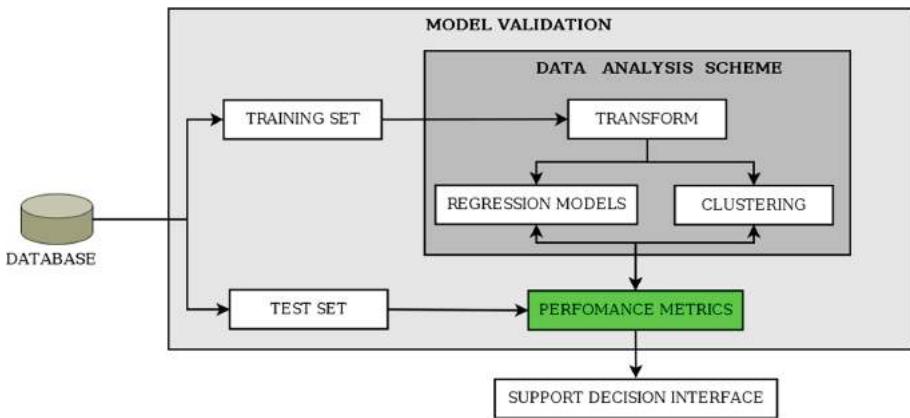


Fig. 1. Data analysis scheme

having some kind of hierarchy in them. With the aim of not saturating the linear regression model with too many coded variables. A new array $\mathbf{X} \in \mathbb{R}^{p \times s}$ has been created. Where $p = 3450$ when eliminating the null data and $s = 12$ when eliminating the municipalities as a characteristic applicable to the model and when increasing columns of the city attribute that are: IBARRA, ANTONIO ANTE, COTACACHI , OTAVALO, PIMAMPIRO and URCUQUÍ . Furthermore, the target variable has been removed from the model to create a vector of characteristics that only stores in this case the total invoiced value.

For its part, to perform the cluster analysis, all the categorical variables must be eliminated to establish similarities in the consumption characteristics. For this reason, a new array called $\mathbf{Z} \in \mathbb{R}^{p \times t}$ is created. Where, $p = 3450$ since the size of the array has not changed. However, $t = 6$ since categorical variables have been removed.

3.4 Regression Models

To generate a model that predicts the total cost of electricity consumption, there are different models using multivariate linear and non-linear regression analysis criteria. In this sense, the most important ones in relation to the literature found are raised as: (i) co-variance matrix, (ii) similarity functions, (iii) frequency table and other. The first criterion (co-variance matrix) is a simple linear model (model 1) based on the equation:

$$y = b_o + \sum_{i=0}^s x_i b_i \quad (1)$$

Where y is the dependent variable, x_i are the independent variables up to the value of $s = 12$ and b_i are the influence parameters of the variables towards the model.

To improve the fit of a linear model, you can square each of its values of the attributes of the matrix and find a multivariate regression (model 2) of the equation:

$$y = b_o + \sum_{j=0}^r \sum_{i=0}^s x_i^j b_i \quad (2)$$

Where the value of $r = 2$ that represents the degree that the linear model has been raised.

Another criterion to consider is based on distances through the k-NN algorithm. The k-NN regression uses the same euclidean distance functions as has the form (model 3):

$$\sqrt[2]{\sum_{i=0}^s (x_i - y_i)} \quad (3)$$

Decision support machines (SVM) can be used as non-linear regression methods since SVM establishes tolerance margins to minimize error, individualizing the hyper-plane that maximizes the prediction margin. This criterion is given by Eq. 4 which can use different kernel methods.

$$y = b_o + \sum_{i=0}^s (x_i - x_i^*).(kernel) \quad (4)$$

It can be seen that the term x_i^* becomes the variable taken to a hyper-plane that has been carried out by a kernel function. For this reason, model 4 uses a *radial* function and model 5 uses a *sigmoid* function.

Finally, algorithms based on frequency tables, such as decision trees and random forests, allow a complex and highly non-linear linear regression model. It is based on Eq. 5.

$$y = \frac{1}{B} + \sum_{i=0}^s f_i \subset x \quad (5)$$

Where $B = 10$ represents the decision trees used and f_i the resulting forecast frequency.

3.5 Clustering

The *k-means* method aims to partition a set of n observations into k groups. Where each value of n belongs to a group of k whose mean value of the distance is the closest.

However, the randomness of the k values can cause different forms of grouping. For this reason, it is necessary to adequately define its value to group the data that have the greatest similarity in their characteristics. Consequently, K-means ++ allows to eliminate this problem by analyzing a set of observations (x_1, x_2, \dots, x_n) , where each observation is a real vector $d - dimensional$ and the *K – means* grouping aims to divide the n observations into $k(\forall n)$ sets

$S = S_1, S_2, \dots, S_k$ to minimize the sum of squares within of the group (WCSS) (the variance). This can be seen in Eq. 6 [10].

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} -\mathbf{x} - \boldsymbol{\mu}_i \xrightarrow{2} = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \text{Var } S_i \quad (6)$$

Where n_{μ_i} is the average of the points in S_i .

Graphically, the variance can be appreciated when using the different k values. For its selection, the value of k should be chosen that no longer exists a high variability in its grouping. In this case $k = 4$. This can be seen in Fig. 2.

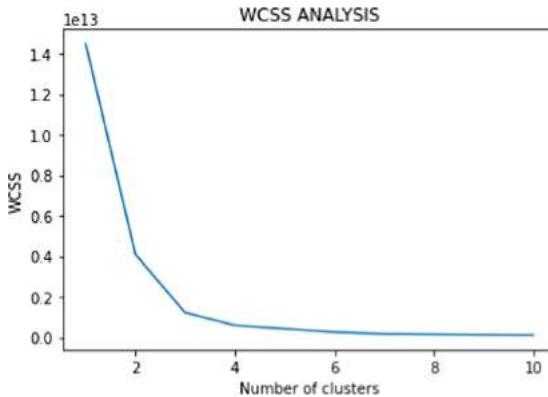


Fig. 2. k analysis by WCSS technique

4 Results

The results are shown with the criteria of: linear regression, grouping and graphical interface.

4.1 Regression Models

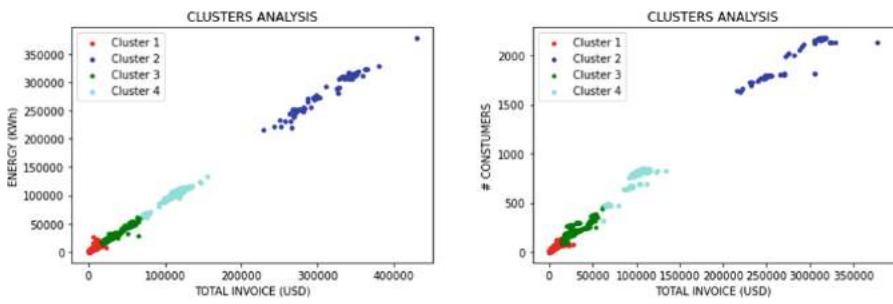
For the selection of the linear regression model that best represents the data set of the matrix $\mathbf{X} \in \mathbb{R}^{p \times s}$. Forecast error tests are performed. To do this, the data matrix has been divided randomized 10 times in training and testing. This is done to train the model and to test the predictions against its response. For this, the accumulated sum of forecast errors (CFE), the mean absolute deviation (MAD), the mean square error (MSE) and the deviation in percentage terms are used (MAE). The Table 1 shows the results obtained by each model proposed with a test set of 750 data. As can be seen, the variability (% MSE and %MAE) is very high in models 1, 2, 3 and 4. For their part, models 5 and 6 have less variability and a forecast error of less than 10%. This result is very acceptable considering the nature of the data.

Table 1. Error type analysis in matrix $\mathbf{X} \in \mathbb{R}^{p \times s}$

Errors type	CFE	%CFE	MAD	MSE	MAE
Model 1	-83007.0	± 7.6	-111.72	152690	14%
Model 2	-82117.9	± 7.52	-110.52	135300	12.43%
Model 3	-83007.0	± 7.6	-111.72	152690	14%
Model 4	-81037.0	± 7.44	-109.06	124017	11.39%
Model 5	-61538.3	± 5.65	-82.82	106848	9.81%
Model 6	-55041.5	± 5.05	-74.08	100914	9.27%

4.2 Clustering

With the result of $k = 4$, proceed to compile the algorithm in the database for grouping by municipality from the matrix $\mathbf{Z} \in \mathbb{R}^{p \times timesteps}$. With this, four types of consumption are defined: *EXCESSIVE*, *HIGH*, *MEDIUM* and *LOW*. With this, the clusters are graphically presented in Fig. 3 the affinity of the data and the distance between them. This shows that the value of $k = 4$ is adequate and represents a 98% variability of the data.



(a) Cluster Analysis by Total Energy Consume (b) Cluster Analysis by Number of Costumers

Fig. 3. Cluster analysis variability

4.3 Interface

The complete data matrix has been loaded into the decision support tool along with the machine learning algorithms for generating reports. With this, personnel of public entities related to electricity generation can observe and manage the results obtained in a friendly way. This can be seen in Fig. 4, where the forecasts of models 5 and 6 are very adequate. In addition, this information can be viewed by cantons or municipalities for greater control and monitoring.

On the other hand, using its two categorical variables and using a treemap as an organization by clusters, all municipalities can be observed with respect

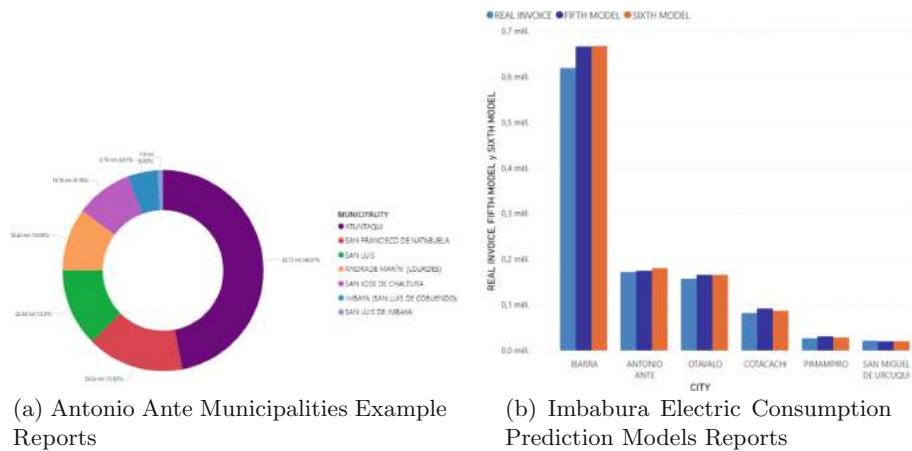


Fig. 4. Support decision interface about consumption predictions

to the group to which they belong and their prediction consumption per month. This information is very important for electrical planning in the province of Imbabura. This is shown in Fig. 5.

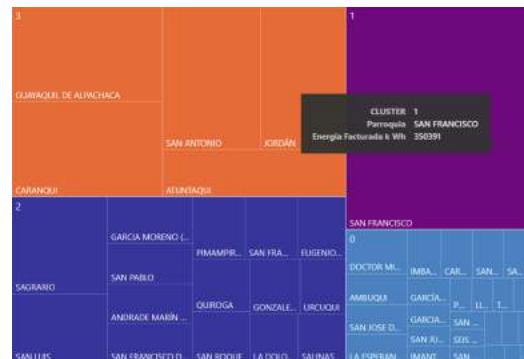


Fig. 5. Cluster Reports by Municipalities: *Excessive Consumption, *High consumption, *Normal Consumption, *Low consumption

Finally, using the global position data of each municipality, each of them can be represented by color with respect to the grouping assigned by the algorithm *EXCESSIVE, HIGH, MEDIUM and LOW*. With this, in each month that the algorithm is recompiled, the change in consumption habits can be observed and alerts can be presented. This is shown in Fig. 6.



Fig. 6. Cluster Reports: *Excessive consumption, *High consumption, *Normal consumption, *Low consumption

5 Conclusions and Future Works

The analysis of electricity consumption by means of automatic learning algorithms allows having robust data analysis for the adequate generation of reports that support the planning and creation of electrical projects that satisfy the increasing electricity demand. In this sense, non-linear regression models allow training models with categorical variables, which are widely used to define certain factors of user characteristics. Consequently, decision support machines and random forests proved to be the most suitable for presenting trends in electricity consumption in the province of Imbabura-Ecuador, organized by cantons and municipalities.

For its part, the *k-means* algorithm with a value of $k = 4$ was adequate to determine consumption similarities by cantons and municipalities. With this, it is possible to observe monthly changes in its trends and alert about them. Finally, the reports for a decision support tool, allow providing agile and efficient information for the electrical planning of a country.

As future work, it is proposed to use these same criteria for all provinces to have a national tool for predicting and grouping electricity consumption.

References

1. Ministerio de Electricidad y Energía Renovable – Ente rector del Sector Eléctrico Ecuatoriano
2. Dmitri, K., Maria, A., Anna, A.: Comparison of regression and neural network approaches to forecast daily power consumption. In: Proceedings - 2016 11th International Forum on Strategic Technology, IFOST 2016, pp. 247–250. Institute of Electrical and Electronics Engineers Inc., March 2017

3. Çetinkaya, Ü., Avcı, E., Bayindir, R.: Time series clustering analysis of energy consumption data. In: 2020 9th International Conference on Renewable Energy Research and Application (ICRERA), pp. 409–413 (2020)
4. Tello Urgilés, D.G.: Universidad Politecnica Salesiana Sede Cuenca Facultad De Ingenierias Carrera De Ingeniería Eléctrica. Technical report
5. Gong, F., Han, N., Li, D., Tian, S.: Trend analysis of building power consumption based on prophet algorithm. In: 2020 Asia Energy and Electrical Engineering Symposium (AEEES), pp. 1002–1006 (2020)
6. Guachimboza-Davalos, J.I., Llanes-Cedeño, E.A., Rubio-Aguiar, R.J., Peralta-Zurita, D.B., Núñez-Barriónuevo, O.F.: Prediction of monthly electricity consumption by cantons in ecuador through neural networks: a case study. In: Botto-Tobar, M., Zamora, W., Larrea Plúa, J., Bazuerto Roldan, J., Santamaría Philco, A. (eds.) Systems and Information Sciences. ICCIS 2020. Advances in Intelligent Systems and Computing, vol. 1273. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-59194-6_3
7. He, L., Song, Q., Shen, J.: K-NN numeric prediction using bagging and instance-relevant combination. In: Proceedings - 2nd International Symposium on Data, Privacy, and E-Commerce, ISDPE 2010, pp. 3–8 (2010)
8. Johnson, B.J., Starke, M.R., Abdelaziz, O.A., Jackson, R.K., Tolbert, L.M.: A method for modeling household occupant behavior to simulate residential energy consumption. In: IEEE PES Innovative Smart Grid Technologies Conference, ISGT 2014 (2014)
9. Liu, Y., Wang, W., Ghadimi, N.: Electricity load forecasting by an improved forecast engine for building level consumers. *Energy* **139**, 18–30 (2017)
10. Núñez-Barriónuevo, O.F., et al.: Clustering analysis of electricity consumption of municipalities in the province of pichincha-ecuador using the k-means algorithm. In: Botto-Tobar, M., et al. (eds.) Systems and Information Sciences, pp. 187–195. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-59194-6_3
11. Narváez, R.P.: Factor de emisión de CO₂ debido a la generación de electricidad en el Ecuador durante el periodo 2001–2014. *Avances en Ciencias e Ingeniería* **7**(2) (2015)
12. Parraga-Alava, J., et al.: A data set for electric power consumption forecasting based on socio-demographic features: data from an area of southern Colombia. *Data Brief* **29**, 105246 (2020)
13. Ministerio de Energía y recursos no renovables De. Plan de mejoramiento de los sistemas de distribución de energía eléctrica (PMD) – Ministerio de Energía y Recursos Naturales no Renovables (2019)
14. Sauhats, A., Varfolomejeva, R., Lmkevics, O., Petrecenko, R., Kunickis, M., Balodis, M.: Analysis and prediction of electricity consumption using smart meter data. In: International Conference on Power Engineering, Energy and Electrical Drives, vol. 2015-September, pp. 17–22. IEEE Computer Society, September 2015
15. Rezaei, S., Sharghi, A., Motalebi, G.: A framework for analysis affecting behavioral factors of residential buildings' occupant in energy consumption. *J. Sustain. Archit. Urban Design* **5**(2), 39–58 (2018)
16. Toapanta-Lema, A., et al.: Regression models comparison for efficiency in electricity consumption in ecuadorian schools: a case of study. In: Botto-Tobar, M., Vizuete, M.Z., Torres-Carrión, P., Montes León, S., Guillermo, V.P., Durakovic, B. (eds.) Applied Technologies, pp. 363–371. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42520-3_29

17. Wang, M., Kou, B., Zhao, X.: Analysis of energy consumption characteristics based on simulation and traction calculation model for the crh electric motor train units. In: 2018 21st International Conference on Electrical Machines and Systems (ICEMS), pp. 2738–2743 (2018)
18. Ye, X., Wu, X., Guo, Y.: Real-time quality prediction of casting billet based on random forest algorithm. In: Proceedings of the 2018 IEEE International Conference on Progress in Informatics and Computing, PIC 2018, pp. 140–143. Institute of Electrical and Electronics Engineers Inc., May 2019
19. Yildiz, B., Bilbao, J.I., Dore, J., Sproul, A.B.: Recent advances in the analysis of residential electricity consumption and applications of smart meter data, December 2017
20. Zhang, X.M., Grolinger, K., Capretz, M.A., Seewald, L.: Forecasting residential energy consumption: single household perspective. In: Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018, pp. 110–117. Institute of Electrical and Electronics Engineers Inc., January 2019
21. Zhao, H.X., Magoulès, F.: A review on the prediction of building energy consumption, August 2012



Street Owl. A Mobile App to Reduce Car Accidents

Mohammad Jabrah, Ahmed Bankher, and Bahjat Fakieh^(✉)

Information Systems Department, King Abdulaziz University, Jeddah, Saudi Arabia
MJabrah0003@stu.kau.edu.sa, BFakieh@kau.edu.sa

Abstract. The irreversible increasing rate of car accidents due to the distraction of using smartphones while driving led this paper to aim for proposing a mobile application that would help to reduce car accidents by limiting the functionality of smartphones during the car movement to the basic needs only, such as calls or maps. The study started by analyzing several applications in the market. Then, the study conducted a survey that was answered by 303 participants revealed the need for a dedicated mobile application to minimize the smartphone distraction while driving. The result shows some interesting information regarding the approximate rate of using smartphones while driving, the common reasons for using smartphones, and some critical reasons that led drivers to avoid touching their mobile phones on the road. Ultimately, the paper proposed a mobile application that is based on the obtained result.

Keywords: Accidents · Driving · Phone calls · Smart phones · Texting · Traffic

1 Introduction

Traffic accidents are considered one of the top 10 causes of death worldwide [1]. In 2018, traffic injuries led to 1.35 million deaths [1]. In that same year in Saudi Arabia, according to the General Authority for Statistics, traffic accident counts reached a high of 352,464, resulting in 30,217 injuries and 6,025 deaths [2]. Out of the 352,464 accidents that took place that year, and according to the Saudi Standards, Metrology and Quality Org. (SASO), as well as their conducted studies and statistics, 161,242 traffic accidents were caused by the use of mobile phones while driving. [3]. These official numbers mean that 46% of the annual traffic accidents in Saudi Arabia were caused by drivers' distraction with mobile phones while driving. This makes mobile phone use the main cause of traffic accidents leading to tragic consequences.

2 Literature Review

2.1 Background and Overview of Related Work

The issue of car accidents in general, and distracted driving, has been one of the major problems of the century. Humanity has suffered heavy losses due to behavior that cause

drivers to lose concentration while driving. Among these dangerous behavior is the use of mobile phones while the vehicle is in motion. The use of a mobile phone while driving exposes not only the driver to the possibility of an accident, but also other passengers in the car, people driving other vehicles on the road, people walking down the street, and even parked cars. As this is considered a major issue, there have been many studies, technological solutions, and governmental attempts in many countries to reduce the scale of this behavior. In this chapter, mobile applications and technologies related to the issue will be discussed, and an overview will be provided to explain the relationship between the mentioned technologies, the issue, and this project.

2.2 Related Mobile Applications

Drive Safe.ly

Developed by iSpeech and suns only in the United States, the application helps solve the problem of texting while driving by reading received SMS messages and emails aloud using iSpeech Text to Speech (TTS). It allows the user to respond by converting the recorded message into a text message and sending it. This option is available only for the Pro option, which is not free of charge [4].

AT&T Drive Mode

This is a free application developed by the American mobile telecommunication company AT&T for Android devices only, as shown in Fig. 1. The application was designed to prevent distracted driving caused by the use of mobile phones by silencing incoming phone calls and alerts so the driver could stay focused. The application starts working after the speed exceeds 15 mph (almost 25 km/h), and it provides a specifically designed interface allowing the driver to access navigation, play music, and key contacts. The application also provides parental alerts, which send a text message to parents if the application is turned off or disabled, or if a new speed dial number is added [5].



Fig. 1. AT&T drive mode interface

Google Assistant Driving Mode

This is an interface made by Google for Android devices. Figure 2 shows that the interface enlarges the size of the icons that represent the most common actions, such as navigation using Google Maps, phone, and media. Additional commands could be customized, such as returning a missed call, resuming previously played media, or navigating to a registered location [6].

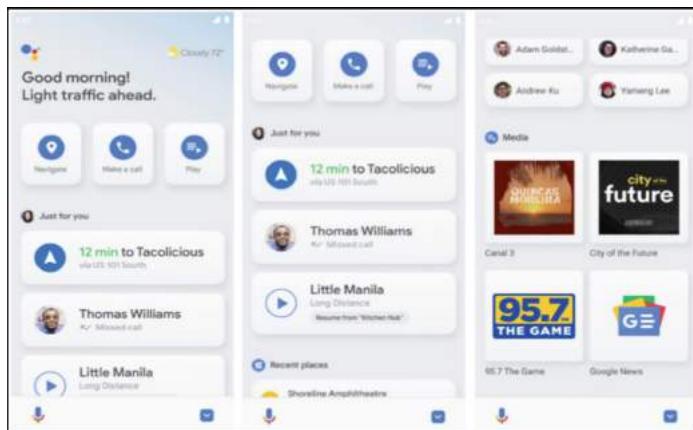


Fig. 2. Interfaces of Google assistant driving mode

Cell Control

The application blocks predefined services and behavior, such as texting, taking pictures and selfies, playing games, and using social media. It can also be used as a monitor to control how teens drive their cars and to check on whether they engage in any bad driving behavior [7].

OneTap

OneTap is an application for Android that helps to control distracted driving during which an individual uses a mobile phone while the vehicle is in motion. The application automatically silences any texts and notifications received and can notify the user when friends or family are currently driving so that the user does not contact them [8]. Two snapshots of OneTap are represented in Fig. 3.

TextLimit

This application is designed to prevent the use of a set of predefined features when the vehicle exceeds a certain speed. The application restores all the prevented features once the vehicle drops back under the predefined speed. The application works symmetrically with the website in need to perform in a complete way [9], as shown in Fig. 4.

Drive Scribe

The application is designed to block incoming texts and calls while the vehicle is in motion. It also notifies drivers when they are traveling at fast speeds [7].

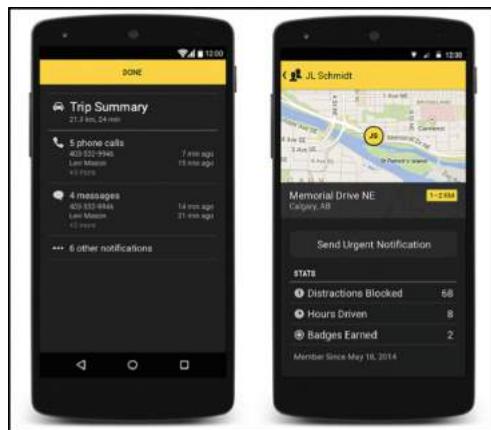


Fig. 3. OneTap application interface

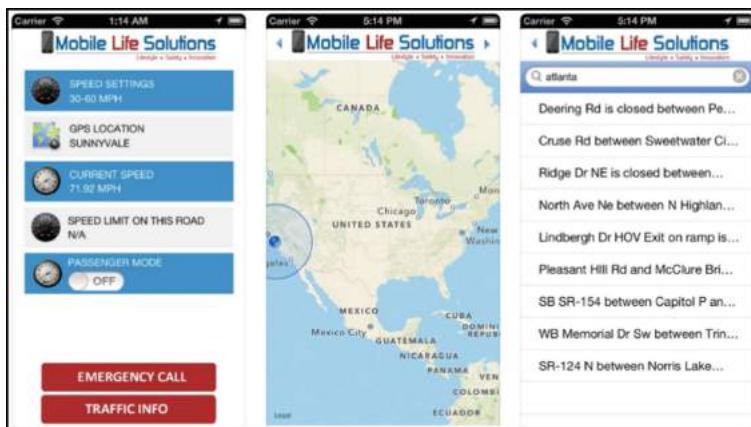


Fig. 4. TextLimit interfaces

True Motion

The True Motion application – as in Fig. 5 – is a family-oriented driving app that allows the user to get a clear picture of the family’s safety. The application allows the user to get details regarding their children’s driving behavior, such as location, phone use, texting, aggressive driving, and speeding [10].

Text Buster

This is a two-part system that includes a physical part to be installed in the car and software for an Android mobile phone. The system costs \$200 and assumes that the driver is using the mobile phone. In that sequence, it sends a notification to the connected mobile phone and blocks texts and calls [11].

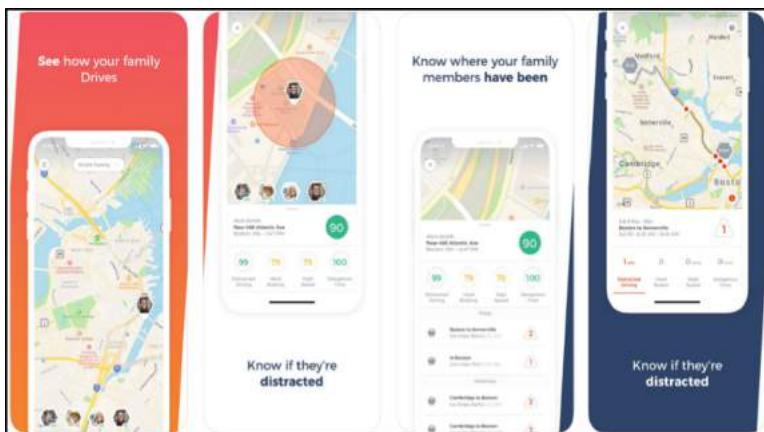


Fig. 5. True motion application interfaces

OMW

This application for iOS allows you to share your location and estimated time of arrival (ETA) with the people whom you are heading to meet. The application aims to reduce mobile usage (e.g., calling or texting other people to inform them of your current location or ETA) while you are driving a vehicle [12].

Car Dashdroid

A replacement for a car’s screen, this converts an Android device into a car home dock screen, as shown in Fig. 6. The application allows the user to reply to text messages such as SMS, WhatsApp, and Telegram without touching the device. The application uses large buttons and icons, as well as full voice commands, and allows the user to create more than 40 shortcuts for easy-access commands.



Fig. 6. Car dashdroid

2.3 Analysis of Related Work

Case Study: Road Traffic Accidents Among Drivers in Abu Dhabi, UAE

The study aims to evaluate and determine the factors related to deaths caused by road traffic accidents. Questionnaires were used to gather data and were deployed in the UK and UAE. Two versions were used: Arabic and English. Six hundred people from the two genders participated in the distributed versions in the city of Abu Dhabi. The study resulted in the identification of several factors causing road traffic accidents.

This research points to the slow progress in traffic accident reduction in UAE and Gulf Cooperation Council (GCC) countries, and to the fact that young adults aged 18 to 35 constitute over 50% of traffic accident deaths in UAE. The first author of the study has experience as a police officer working with the traffic department in Abu Dhabi. Based on that experience, it was noted that the number of traffic violations is increasing, as is the number of traffic accidents involving injured young adults.

The study also explains how much traffic accidents cost different countries in terms of their Gross National Product (GNP); they cost 2% of the GNP in high-income countries, 1.5% in middle-income countries, and 1% in low-income countries. In general, road traffic accidents cost the global community about US\$518 billion. The study points to efforts made by the government of the United Arab Emirates to reduce traffic accidents and make roads safer for all drivers and passengers. The study explains how males are more likely to adopt risky driving behavior and to be involved in car crashes as compared to female drivers.

Quantitative data methods were used through questionnaire surveys, in which drivers were asked to identify “relevant factors that contribute to the traffic accidents in Abu Dhabi.”

The most commonly cited factors were:

- A significant percentage of drivers from both genders do not wear seat belts regularly
- Use of a mobile phone while driving

- Drinking alcohol and driving
- Aggressive driving behavior
- Unsafe driving behavior
- Lack of correct targeting of traffic campaigns.

The study concluded by highlighting the mentioned problems identified as major factors causing traffic accidents and suggested developing safety measures on roads, as well as developing educational programs and campaigns targeting the right groups so that results can be noticed [13].

Mobile Phone Usage and the Growing Issue of Distracted Drivers

A report by the World Health Organization (WHO) focuses, in an accurate and detailed way, on the issue of mobile phone use while driving and aims to raise awareness of distracted driving. The report states that mobile phone use causes drivers to take their eyes off the road and their hands off the steering wheel, which has the biggest impact on driving behavior. The listed body of evidence includes longer reaction times for acts related to such things as identifying traffic signals and braking, the ability to keep the vehicle in the correct lane, shorter following distances, and the condition in which the driver is not aware of the surrounding conditions. The report indicates that the rapid growth of text messaging between drivers is a major problem that requires attention. Nevertheless, young drivers are thought to use mobile phones while driving more than older drivers do, which makes them more vulnerable to the effects of this behavior due to their lack of experience in reacting to road conditions.

The report also indicates that different studies suggest that crash risk related to the use of mobile phones appears to be approximately 4 times higher, and that this risk is similar for both hand-held and hands-free devices, suggesting that the reason for this increased risk is not the act of holding the phone, but the loss of concentration while one is involved in a phone conversation.

It is stated that, annually, 1.3 million people die and another 50 million are injured due to traffic accidents, alongside the significant casualties and economic losses that accompany the emotional toll.

According to the WHO's report, driver distraction can be split into four types:

- Visual: "looking away from the road for a non-driving-related task"
- Cognitive: "reflecting on a subject of conversation as a result of talking on the phone – rather than analyzing the road situation"
- Physical: "when the driver holds or operates a device rather than steering with both hands, or dialing on a mobile phone or leaning over to tune a radio that may lead to rotating the steering wheel"
- Auditory: "responding to a ringing mobile phone, or if a device is turned up so loud that it masks other sounds, such as ambulance sirens"

The report also points to the lack of official and registered data regarding traffic accidents in relation to driver distraction, especially when crash details are registered by the police and no driver distraction data are noted. This makes it difficult for researchers and people who are seeking solutions to the issue.

There are two sources of driver distraction: in-vehicle (interior) distractions, which occur when the driver plays with the radio or mobile phone or changes music tracks, and external distractions, which occur when a driver's eyes are taken off the road while the driver is looking at buildings.

The report included a selection of studies suggesting that distraction is an important contributor to traffic accidents. These are quoted below:

"An Australian study examined the role of self-reported driver distraction in serious road crashes resulting in hospital attendance and found that distraction was a contributing factor in 14% of crashes".

"In New Zealand, research suggests that distraction contributes to at least 10% of fatal crashes and 9% of injury crashes, with an estimated social cost of NZ\$ 413 million in 2008 (approximately US\$ 311 million). Young people are particularly likely to be involved in crashes relating to driver distraction".

"Insurance companies in Colombia reported that 9% of all road traffic crashes were caused by distracted drivers in 2006. Of all cases where pedestrians were hit by cars, 21% were caused by distracted drivers".

"In Spain, an estimated 37% of road traffic crashes in 2008 were related to driver distraction".

"In the Netherlands, the use of mobile phones while driving was responsible for 8.3% of the total number of dead and injured victims in 2004".

"In Canada, national data from 2003–2007 show that 10.7% of all drivers killed or injured were distracted at the time of the crash".

"In the United States, driver distraction as a result of sources internal to the vehicle was estimated to be responsible for 11% of national crashes that occurred between 2005 and 2007, although a smaller study involving 100 drivers found that driver involvement in secondary tasks contributed to 22% of all near crashes and crashes (25, 26). In 2008, driver distraction was reported to have been involved in 16% of all fatal crashes in the United States".

"In Great Britain, distraction was cited as a contributory factor in 2% of reported crashes. The difficulty for the reporting officer to identify driver distraction is likely to have led to an under-representation of this proportion, which is considered a subjective judgement by the police. In addition, contributory factors are disclosable in court and police officers would require some supporting evidence before reporting certain data, a factor likely to lead to an under-representation of the problem".

The report suggests that technological systems could be used within vehicles to prevent distraction in general, such as lane-changing warning features.

The report concludes with an acknowledgment of the importance of mobile phones in improving communications, while also citing mobile phones as a major cause of distracted driving. It also urges governments to improve safety measures and current gains, in order to take advantage of the developing technologies and help mitigate the issue in the future [14].

3 Research Problem

Traffic accidents are considered among the main killers of children and young adults worldwide, and mobile phones are considered one of the top 10 causes of traffic accidents

[15]. In the Kingdom of Saudi Arabia specifically, distraction caused by the use of mobile phones while driving is responsible for almost half of the recorded traffic accidents, which result in a high number of injuries, deaths, and damage to or loss of property. Thus, this research aims to address the following:

1. Highlighting the impact of mobile phone use while driving on traffic accident rates in Saudi Arabia.
2. Studying the issue of traffic accidents caused by distraction resulting from the use of mobile phones from both social and scientific perspectives and in relation to previous studies done in the same field.
3. Providing a mobile application to help minimize the likelihood of accidents.

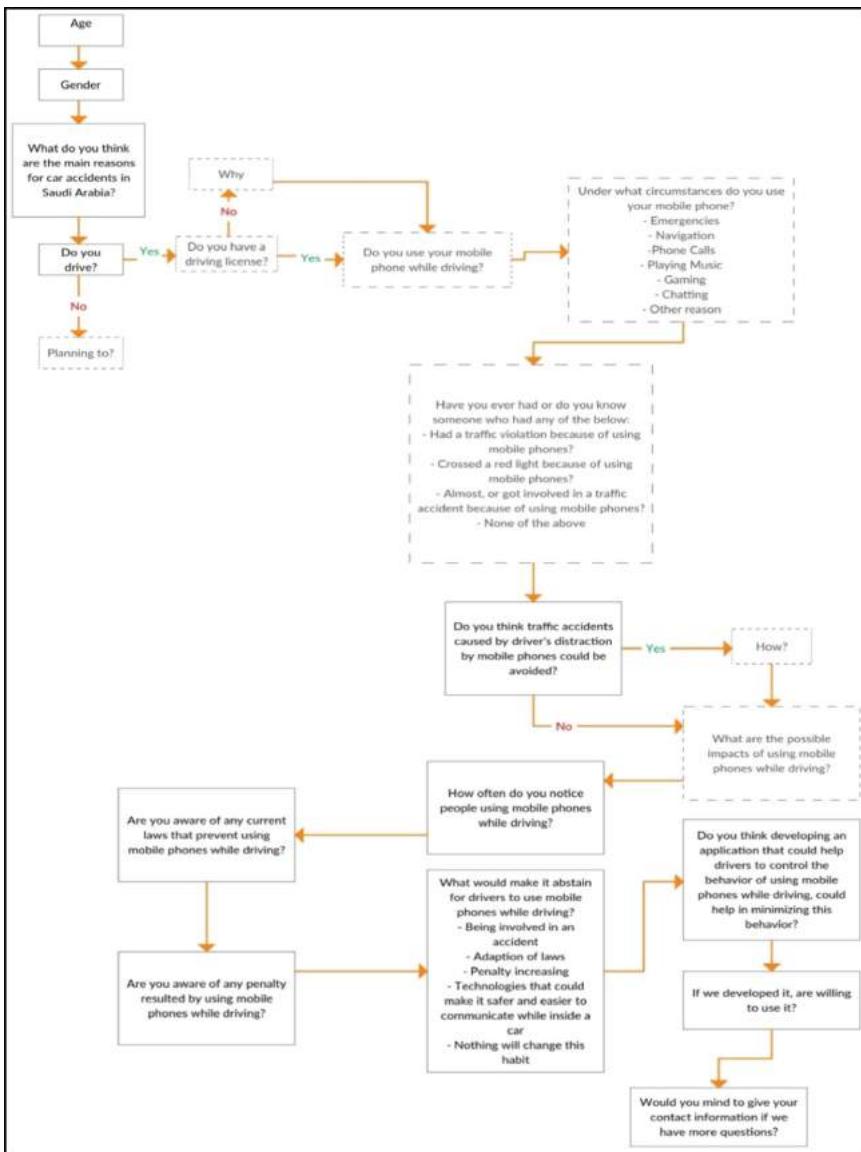
4 Methodology

This project is divided into two main phases. In the first phase, a questionnaire will be created to gather data from a specific community segment, i.e., drivers between 18 and 64. Then, the collected data will be analyzed to obtain an understanding of mobile phone use while driving and the accident ratio related to mobile phone use. A specialized software application such as QlikView will be used to analyze and visualize data. We will develop a mobile application using specialized software such as Android Studio or alternatives. A proper database will be built while progress is being made during the project's timeline, which will be built using MySQL or similar tools.

5 Analysis

5.1 Data Collection

A study questionnaire was conducted as part of the project. The purpose of the study is to measure the scale of society regarding the issue of distracted driving due to the use of mobile phones, and how widespread the issue is. The questionnaire was conducted online in the Arabic and English languages. Figure 7 presents the flow of the conducted study.

**Fig. 7.** The survey structure

5.2 Observation Description

The questions were carefully worded to fulfill the needed output of the questionnaire. They were chosen to provide an overview of society's reactions and the scale of the issue, as well as what solutions are being offered to solve it, for purposes of comparison with the proposed solution.

5.3 Observation Analysis

Q1: Gender

This question measures the number of respondents based on their gender, as in Fig. 8. In total, 142 respondents were female, with a percentage of 47.02%, while 161 were male, with a percentage of 52.98%.

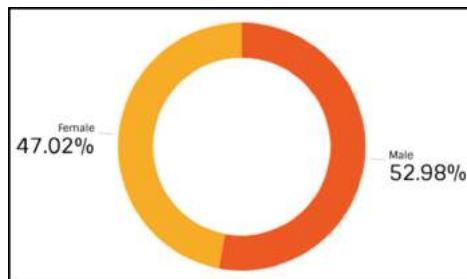


Fig. 8. The gender of the participants

Q2: Age

This question indicates the number of participants in relation to different age groups. Figure 9 shows that 18–25 represents the group of university students, 26–30 represents new graduates and master's degree students, and the 31–40 and 40+ groups represent older adults.

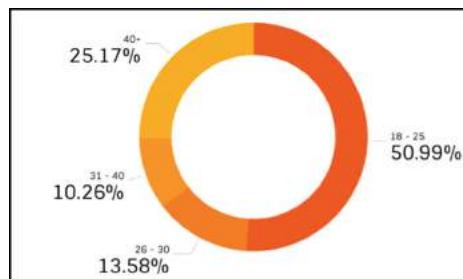


Fig. 9. The age range of the participants

Q3: What do You Think are the Main Reasons for Car Accidents in Saudi Arabia?

This question seeks participants' opinions regarding the main reasons for car accidents in Saudi Arabia. Figure 10 indicates that distracted driving, which includes the use of a mobile phone while driving, represents the highest percentage, at 44%. Reckless driving was second, at 25.67%. Speeding came in third, at 15.67%.

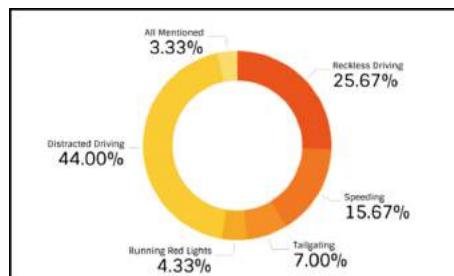


Fig. 10. The main reasons for accidents, based on society's opinion

Q4: Do You Drive a Motor Vehicle?

This question was used as a filter to differentiate between participants who drove a vehicle and those who did not, so that people who drove could be targeted with more specific questions. Figure 11 shows that 64.90% of participants drove a vehicle, while 35.10% did not.

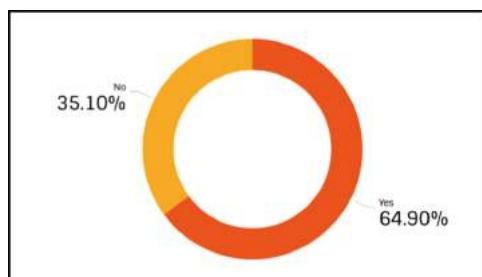


Fig. 11. Rate of participants who drive

Q5: Are You Considering Driving in the Future?

This question was asked of people who answered "no" to the previous question, simply to

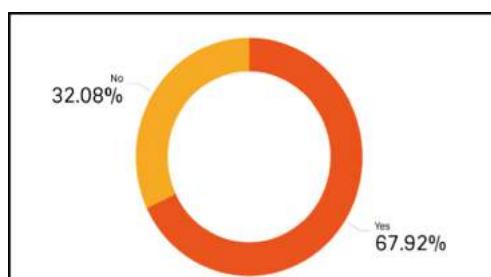


Fig. 12. The rate of non-driving participants who are considering driving

measure their desire to drive. The result in Fig. 12 indicates that 67.92% were considering driving and 32.08 were not.

Q6: Do You Have a Driving License?

This question was asked only of participants who drove, to measure the percentage of legal and illegal drivers. The result in Fig. 13 shows that 94.39% of the participants had a driving license, while 5.61% did not.

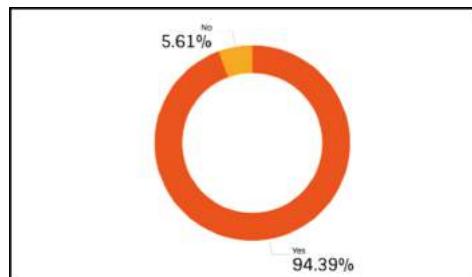


Fig. 13. Rate of driving-participants who have a driving license

Q7: What is the Reason for not Having a Driving License Yet? You can Select More Than One Answer

This question was asked only of people who did not hold a driving license, to determine why they did not yet have one. A simple representation of the result is shown in Fig. 14.

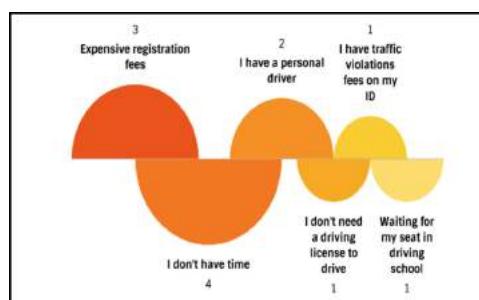


Fig. 14. The reasons for driving without a driving license

Q8: Do You Use Your Mobile Phone While Driving?

This question sought the percentage of participants who used their mobile phones while driving. The results in Fig. 15 indicate that 62.24% (over half) said yes, while 37.76% said no.

Q9: Under Which Circumstances do You Use Your Mobile Phone While Driving? You can Select More Than One Answer

This question was asked to determine the reasons why drivers used their mobile phones, which helps in understanding the behavior. The results, as revealed in Fig. 16, indicated that making phone calls was the top reason. Navigation and Google Maps came in second, while “for emergencies only” was third. Playing music was the fourth reason, while chatting was the fifth. Social networks came in sixth, while gaming came in last, at seventh. Seventeen people said that they did not use mobile phones while driving.

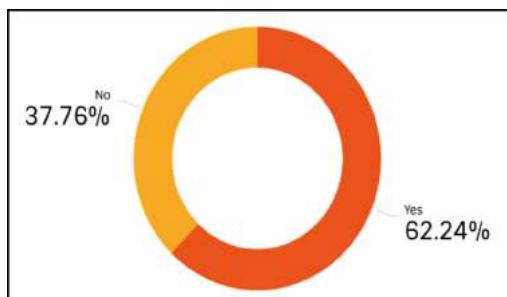


Fig. 15. The rate of people who use a mobile phone while driving

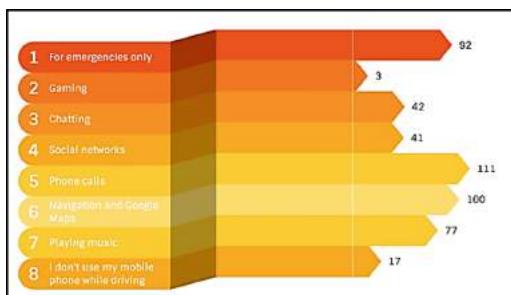


Fig. 16. The reasons why people use their mobile phones while driving

Q10: Have You Ever Done Any of the Below, or do You know Someone Who had an Experience With Any of the Below? You can Select More Than One Answer

This question was asked to determine the scale of the impact of using mobile phones on lives, property loss or damage, traffic violations, and other circumstances. Figure 17 shows the results as the following: 41.3% of the participants had a traffic violation themselves or knew someone who did, because of the use of mobile phones; 28.2% of the participants had gotten into a traffic accident or almost gotten into one, or knew someone who did or almost did, because of the use of mobile phones; and 7.3% of the participants had run a red light themselves or knew someone who did because of the use of mobile phones. Meanwhile, 23.2% of the participants had not encountered any of the abovementioned circumstances.

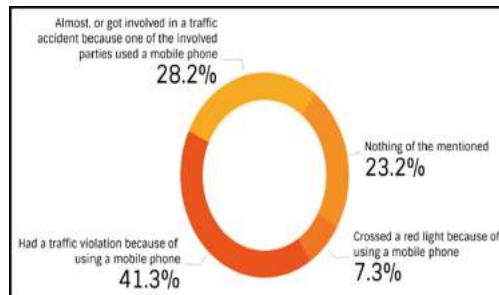


Fig. 17. The rate of people who have experienced the results of distracted driving

Q11: Do You Think Traffic Accidents Caused by Drivers' Distraction while Using Mobile phones Could be Avoided?

This question was asked to determine society's perspective regarding the possibility of avoiding traffic accidents caused by mobile phone use. The results in Fig. 18 show that 82.14% of the participants agreed that such accidents could be avoided, while 17.86% did not.

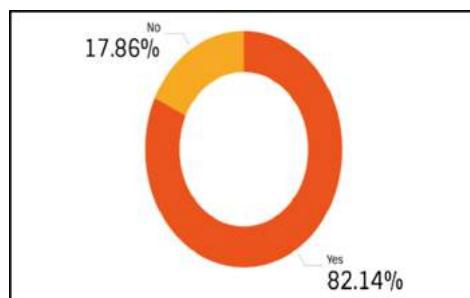


Fig. 18. The rate of people who think that distracted driving could be avoided

Q12: How Often do You Notice People Using Their Mobile Phones While Driving?

This question was asked to determine how frequently participants noticed people using their mobile phones while driving, which could help in measuring the scale of the problem. The question used the scale method, which is represented by five degrees, as in Fig. 19. The first choice was "I don't notice them" and the fifth choice was "I always notice a lot of them". The fifth and fourth options were each chosen 73 times, while the third option was chosen 46 times. The second option was chosen four times. No participant selected the first option.

Q13: Are You Aware of Any Current Laws that Prohibit the Use of Mobile Phones While Driving in Saudi Arabia?

This question was asked to measure society's awareness regarding the laws that prevent the use of mobile phones while driving in Saudi Arabia. Figure 20 shows that 95.41% of participants were aware of the laws, while 4.59% were not.

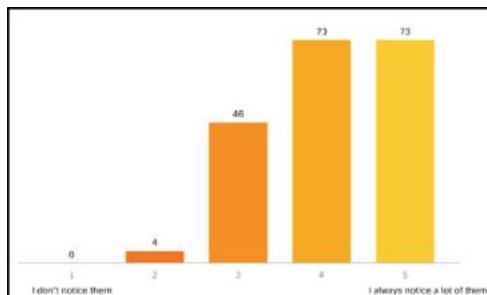


Fig. 19. The rate of people who notice drivers using mobile phones

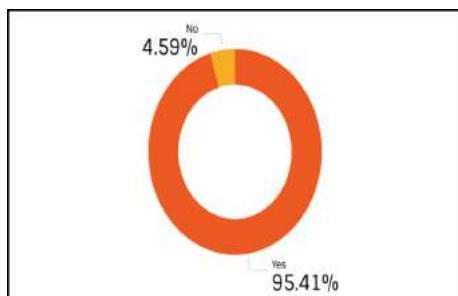


Fig. 20. The rate of awareness of laws

Q14: Are you Aware of any Penalty as a Result of the Use of Mobile Phones While Driving in Saudi Arabia?

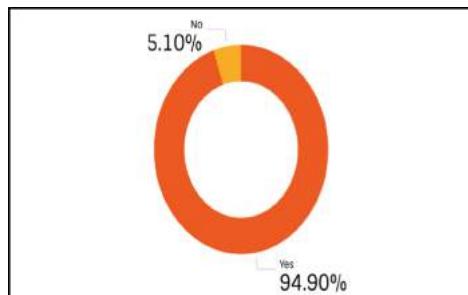


Fig. 21. The rate of awareness of penalties

This question was asked to measure society's awareness regarding the penalties resulting from traffic violations related to the use of mobile phones while driving in Saudi Arabia. Figure 21 indicates that 94.90% of participants were aware of the penalties, while 5.10% were not.

Q15: What Would Make Drivers Abstain From Using Their Mobile Phones while Driving? You can Select More Than One Answer

This question was asked to determine society's opinion regarding the factors that would cause drivers to stop using their mobile phones when driving. Figure 22 shows that "Technology" was the most-chosen option 131 times. "Penalty increasing" came in second with 100 votes, while the option "After being involved in an accident" was third with 88 votes. "Adaption to laws" came in fourth with 73 votes, while there were 11 votes for "nothing will change this habit".

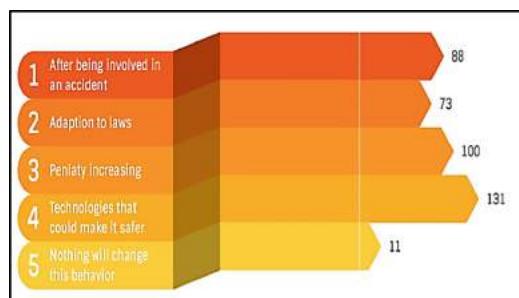


Fig. 22. What would stop people from using their mobiles while driving

Q16: Do you think that Developing a Mobile Application that Could Help Drivers Control the Behavior of Using Mobile Phones While Driving could Help Minimize the Scale of this Behavior?

This question was asked to measure society's opinion regarding the provision of a mobile

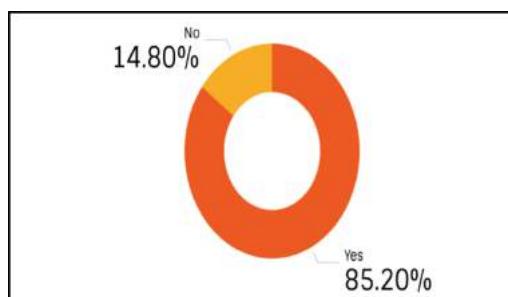


Fig. 23. The rate of people who think a mobile app can help solve the issue

application as a solution to this problem. Figure 23 shows that 85.20% of participants agreed with the idea, while 14.80% did not.

Q17: If we Developed such an App, Would You be Willing to use it?

This question was asked to measure participants' acceptance and to determine whether there was space for such an idea. Figure 24 shows that 81.63% of participants said that they would be willing to use such an app, while 18.37% were not willing to do so.

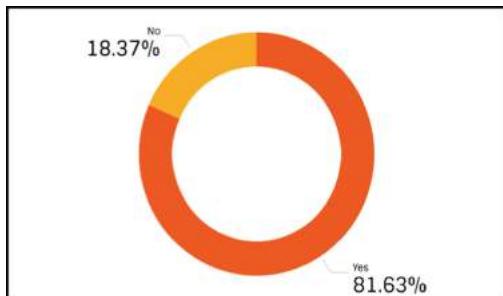


Fig. 24. The rate of people who are willing to use the proposed solution

Gender-Based Usage of Mobile Phones While Driving

Based on the gathered results and after analysis, Fig. 25 shows that 72.5% of females from among the entire count of female participants did not use their mobile phones while driving, while 27.5% of females did. On the other hand, it was found that 28.8% of males from among the entire count of male participants did not use their mobile phones while driving, while 71.2% of males did use their mobile phones. The two percentages show that men use their mobile phones more frequently than women do.

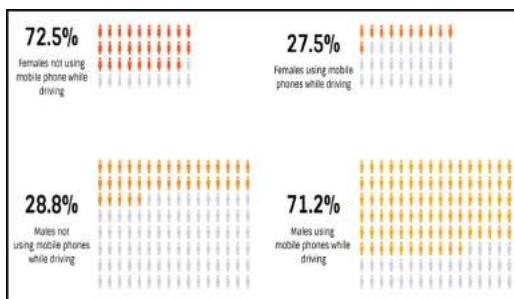


Fig. 25. The relation between each gender and the usage of mobile phones while driving

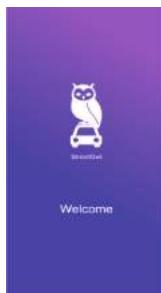
5.4 Discussion

The results of the performed study indicate that men use their mobile phones more than women do, and that women are more committed to the laws. Society has a high

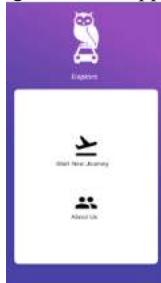
acceptance level for the proposed solution, which is a mobile application to help drivers control the behavior of using mobile phones while driving. Society's awareness of the laws and penalties regarding the traffic violation of using mobile phones while driving is prospectively high; nevertheless, there should be more advertisements and campaigns to educate current drivers and upcoming generations about the issue and why they should control this behavior, as some drivers do not understand what they are risking when they use their mobile phones while driving.

6 Interface Design

To satisfy the users and deliver a well-designed product, the interface design should follow the principles of human-computer interaction (HCI). The application was named



A: The welcome page once the application is launched



B: The user chooses to start a new journey or view the developer's info



C: New journey interface

Fig. 26. Street owl app interface

“Street Owl” and was designed to have a consistent appearance across the system, as shown in Fig. 26.

The user can understand their current location while using the application. Navigation methods such as buttons, as well as other methods and elements, will be used to enable the communicating of information and the initiating of actions.

Primary content—such as numbers, text, graphics, interactive elements, etc.—should be available on-screen, as well as easy to notice and understand. Scrolling, zooming, and other behavior should be enabled. Many types of controls—such as buttons, text fields, and progress indicators—will be used to make the experience easy for the user.

Images and icons will be added using the standards of user experience (UX), in addition to the fonts’ sizes and colors, backgrounds colors, and other elements.

7 Usability Testing

The usability testing went through five stages, which were:

1. Alpha Testing. This is the most common type of testing used in the software industry. The objective of this type of testing is to identify all possible issues or defects before releasing the product into the market or to the user. Alpha testing is carried out at the end of the software development phase but before the beta testing phase. Minor design changes may be made as a result of such testing. Alpha testing is conducted at the developer’s site. An in-house virtual user environment can be created for this type of testing.
2. Accessibility Testing. The aim of accessibility testing is to determine whether the software or application is accessible for disabled people. Here, disability means deafness, color blindness, mental disability, blindness, old age, and other disabilities. Various checks are performed, such as font size for the visually disabled, color and contrast for those with color blindness, etc.
3. Beta Testing: Beta testing is a formal type of software testing carried out by the customer. It is performed in the real environment before the product is released to the market for the actual end-users. Beta testing is carried out to ensure that there are no major failures in the software or product and that it satisfies the business requirements from an end-user perspective. Beta testing is successful when the customer accepts the software. Usually, this testing is done by end-users or others. It is the final testing done before a product is released for commercial purposes. Usually, the beta version of the software or product released is limited to a certain number of users in a specific area. Thus, the end-user actually uses the software and shares their feedback with the company. The company then takes necessary action before releasing the software worldwide.
4. Black Box: Testing internal system design is not considered in this type of testing. Tests are based on the requirements and functionality. Detailed information about the advantages, disadvantages, and types of black box testing can be seen here.
5. Gorilla Testing: Gorilla testing is a testing type performed by a tester and sometimes by the developer as well. In gorilla testing, one module or functionality in the module is tested thoroughly and heavily. The objective of this testing is to check the robustness of the application.

8 Conclusion

Driving and using a mobile phone simultaneously could expose a driver, passengers, people walking on the side of the road, and even property to losses and damages. This behavior is considered one of the main causes of death worldwide, especially in Saudi Arabia. We believe that using technology to help monitor, control, and assist drivers in maintaining safe driving behavior is important. Street Owl could help people control themselves and monitor their behavior if they find it difficult to do so without help. More awareness should be spread—especially in large cities such as Jeddah, Riyadh, Makkah, and Khobar—regarding the results of using anything that would lead to the distraction of the driver. More efforts should be made by both the authorities and the people of Saudi Arabia to make our streets safer for everyone.

References

1. WHO. Global status report on road safety 2018. 2018 26 Sep 2019. https://www.who.int/violence_injury_prevention/road_safety_status/2018/en/
2. Statistics, G.A.f.: Statistical Yearbook of 2018 | Chapter 14 | Transportation. 2018 [cited 2019 25 Sep 2019]; Official Traffic Accidents Statitics Published by GAS]. <https://www.stats.gov.sa/en/1020>
3. SASO. SASO: 161,242 Traffic Accidents Take Place Annually Due to the Use of Mobile Phones While Driving. 2019 [cited 2019 25 Sep 2019]. https://www.saso.gov.sa/en/mediacenter/news/Pages/saso_news_973.aspx
4. Oltzik, J.: What's Needed for Cloud Computing? Enterprise Strategy Group, 10 (2010)
5. AT&T. AT&T DriveMode. 2018 [cited 2019 25 Oct 2019]. https://about.att.com/sites/it_can_wait_drivemode
6. Kerns, T.: Google is getting rid of Android Auto's smartphone UI. 2019 [cited 2019 25 Oct 2019]. <https://www.androidpolice.com/2019/07/30/android-auto-app-going-away-assistant-driving-mode/>
7. Winkler, V.J.R.: Securing the Cloud. 2011: Syngress
8. Buyya, R., et al.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Futur. Gener. Comput. Syst.* **25**(6), 599–616 (2009)
9. MobileLifeSolutions. Text Limit. 2013 [cited 2019 26 Oct 2019]. <https://apps.apple.com/us/app/text-limit/id612385844>
10. TrueMotion. TrueMotion Family Safe Driving. 2016 [cited 2019 25 Oct 2019]. <https://apps.apple.com/us/app/truemotion-family-safe-driving/id1121316964>
11. Goodwin, A. TextBuster. 2013 [cited 2019 26 Oct 2019]. <https://www.cnet.com/pictures/textbuster/2/>
12. Jorissen, K., Vila, F.D., Rehr, J.J.: A high performance scientific cloud computing environment for materials simulations. *Comput. Phys. Commun.* **183**(9), 1911–1919 (2012)
13. Hammoudi, A., Karani, G., Littlewood, J.: Road traffic accidents among drivers in Abu Dhabi, United Arab Emirates. *J. Traffic Logistics Eng.* **2** (2014)
14. WHO. Mobile phone use: a growing problem of driver distraction. 2014 [cited 2019 26 Oct 2019]. https://www.who.int/violence_injury_prevention/publications/road_traffic/distra cted_driving/en/
15. WHO. The top 10 causes of death. 2018 [cited 2019 5 Oct 2019]. <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death#targetText=Ischaemic%20heart%20dise ase%20and%20stroke,in%20the%20last%2015%20years>



Low-Cost Digital Twin Framework for 3D Modeling of Homogenous Urban Zones

Emad Felemban¹(✉), Abdur Rahman Muhammad Abdul Majid^{2,3},
Faizan Ur Rehman^{2,3}, and Ahmed Lbath²

¹ Computer Engineering Department, College of Computing and Information Systems, Umm Al-Qura University, Mecca, Saudi Arabia
eafelemban@uqu.edu.sa

² Laboratoire d'Informatique de Grenoble, University of Grenoble Alpes, Grenoble, France
fsrehman@uqu.edu.sa, ahmed.lbath@univ-grenoble-alpes.fr

³ Institute of Consulting and Research Studies, Umm Al-Qura University, Mecca, Saudi Arabia

Abstract. Smart cities have been on the rise since the last decade. These cities can only be effective and sustainable when acquiring error-free information from multiple aspects and sources. Prone to errors and mistakes, traditional tools and observation methods require new assessment strategies to provide a clear perspective in simulation and a systematic evaluation for decision-making to be aligned with Industry 4.0. Digital Twin can offer an efficient solution to this problem under the cyber-physical system entity by creating a virtual space identical to the physical region and providing predictive information on the city's current state. Such systems offer accurate representation in a virtual environment while taking input from the real world and simulating them for future predictions. These near-identical systems are constructed at a very high cost, and the cost increases as more intricate details are added to the environment. This paper presents selective technologies that can potentially contribute to developing a low-cost intelligent environment and smarter urban management framework. It looks into the analysis of the impact of different scenarios on a public event in three dimensions that will be crucial to the decision and policymaker before a plan is approved. The proposed framework will be able to guide present and upcoming potential solutions against the administrative challenges.

Keywords: Digital twin · Low-cost · 3D Modeling · Data collection · Smart city

1 Introduction

Traffic, safety, and energy are fields in which various cities have increased consumption. Many problems are occurring as a cause, such as traffic congestion and water shortage. Europe, Japan, and the United States are some of those regions that are investing heavily in the creation of smart cities and their related technologies. China is also using intelligent cities for urban innovation. Smart cities and their related industries are expected to grow to \$1.5 trillion in 2020 [1].

In such a smart city, computing technologies and information and communication technologies converge and are combined as Digital Twin to provide a core joint platform. In other words, real-time simulation, artificial intelligence, IoT, spatial information is utilized to build a converged digital twin platform. Information necessary for city operations is integrated and virtual to modify the real world and solve various urban problems. The core infrastructure to implement a smart city platform is spatial information to decode the problem, such as accurate status analysis and results within the spatial infrastructure, and its information is essential for the city's transformation to a smart city. This spatial information can be made visible and efficient using aerial Lidar surveys, Ground lidar surveys, vehicle lidar surveys, and multi-directional aerial surveys [2]. Research has been carried out to generate actual 3D spatial information and its construction based on various sensors. Recently, the demand for the latest low-cost technology for useful 3D spatial information is increasing.

Therefore, the purpose of this study is to provide a low-cost and high-quality 3D framework. Building a 3D model of the Mina area in Mecca, Saudi Arabia using multiple technologies and sensors to acquire information will be used. Also, as a digital twin model, the built model's quality and efficiency analysis will be performed, and the efficacy and possibility of application in the application field will be discussed.

For the rest of the paper, the authors discuss the current research trends related to digital twin technologies. The framework design methodology is discussed in detail, focusing on each aspect of the Digital twin framework, followed by a brief implementation section. The paper is then concluded by discussing the framework and the limitations of such a low-cost alternative, followed by concluding remarks and intended future work.

2 Research Trends

The construction of a 3D digital model for a high-density, verticalized urban space is essential not only for constructing the spatial information base of smart cities but also for the construction of three-dimensional intelligence and cyber-physical systems. It is regarded as an element, and studies on manual, semi-automated, and automated have been actively conducted since the 1990s based on various sensors geared towards an Industry 4.0 revolution. As the current policy has led the drone industry to the fourth industrial revolution, research is actively underway to take advantage of developing a low-altitude, high-precision, and low-cost 3D spatial model. Such enhancement was otherwise not possible to obtain and generate from the existing operating aerial surveying base.

Looking at the study of accuracy analysis of drone photogrammetry, Lee et al. [3] produced orthogonal images and digital surface models with 4cm/pixel spatial resolution using fixed-wing and low-cost rotary-wing drones for topographic surveying of civil engineering construction sites. The root-mean-square error was analyzed to be around 10 cm in the x, y, and z directions compared with the outcome. NO et al. [4] proposed an underground utility mapping using ground control points the reduced the error bound of the acquired data.

Guisado-Pintado et al. [5] mounted a general camera on a rotorcraft drone. They obtained an accuracy of 9cm horizontal (x,y) error and 7.7cm vertical (z) error as a

result of error analysis on the inspection point in a study to evaluate the effectiveness of the drone photogrammetry method for production, the result of verifying the digital topographic map produced by drone photogrammetry by the ground survey method results in a plane error of an average of 4cm and a maximum of 8cm, and an elevation error of an average of 11cm and a maximum of 31cm.

NO et al. [4] further compared the ground survey coordinates of the buried pipe in the experimental area, and the 17 processing results of the drone photographed data 7 times to create an underground facility map using a drone. It was suggested that excess was seen in only two processing.

As a result of consideration through preliminary research, the accuracy of drone photogrammetry varies depending on the shooting height, camera resolution, and redundancy at the time of drone photography. However, the plane error is within 10cm on average. It shows high accuracy compared to the standard deviation within 14cm and the maximum value within 28cm, which is the error limit of the plane reference point.

While comparing simulation modeling and digital twin, Krasikov et al. [6] identified the main difference in the data flow structure of the model. They proposed that the loop between the physical space and the virtual should be automated and continuous.

3 Framework Design Methodology

As shown in Fig. 1, given an existing physical product, in general, it takes several steps to create a fully functional and responsive digital twin. It should be made clear that in practice, manufacturers and developers do not strictly follow such a sequence while developing a Digital Twin model [7]. Discussed below are generalized steps that were taken while developing the economical yet practical digital twin framework.

3.1 Spatial Data Acquisition

This step deals with the process of spatial data acquisition. The goal is to collect and acquire quickly in an economical manner. The initial step in acquiring data was an onsite inspection of camps where they were measured using tools. The pedestrian bridge details were acquired for a vendor that provided the entire floors and ramps of the bridge realistically. Intel's RealSense™ Lidar Camera¹ is also used to collect data. Such low-cost Lidar scanning devices can help in providing realistic textures and measurements for validation. Other standard acquisition methods allow users to mount a scanning module on a moving vehicle that gathers information from the surrounding in the form of cloud points, which makes the data more accurate, however that is not a cost-effective solution as the module have exorbitant prices that does not justify the use case it is being used for scanning [8].

3.2 Building a 3D Environment

This process is being enabled using the finalized computer-aided drawing (CAD) constructed from the measurement from the previous steps. Since the Mina area structures

¹ Intel® RealSense™ LiDAR Camera L515 - [link](#).

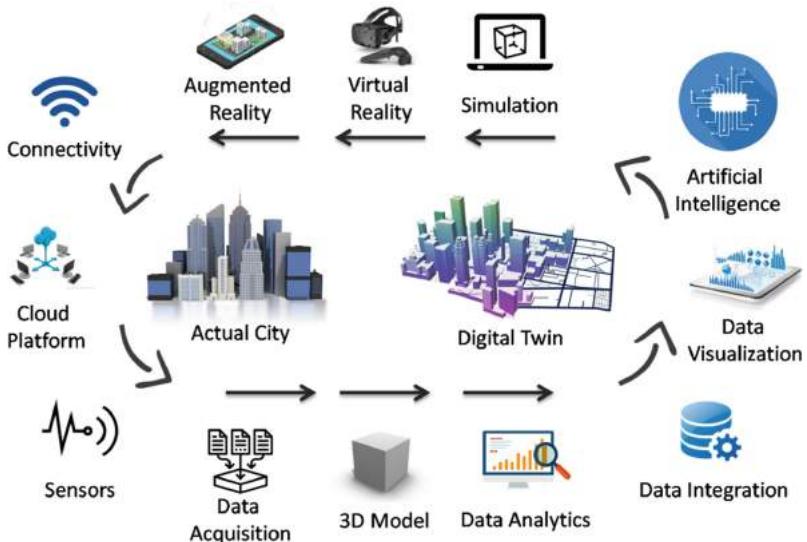


Fig. 1. Digital twin framework lifecycle

are mostly homogenous (see Fig. 2), these buildings are quickly drawn. These measurements were also provided to the graphics designer to perform the extrusion process for surfaces in the Blender² that allows us to model the camps in the Mina area effectively. Blender is also a freely available 3D creation suite that aids in the cost-effective solution provided by the authors. 3D modeling is then performed on these CAD drawings to depict the camp and the bridge structures. The virtual environment includes three aspects, elements, behaviors, and rules [9]. At the level of elements, the virtual space model mainly includes the physical environment's geometric model, dynamic or static entities, and the environment's user or an administrator. At the level of behaviors, the authors analyze the behavior of environments and users and focus on analyzing the environment and user interaction generated by the behavior and modeling. The rules level mainly includes the evaluation, optimization, and forecasting models established following the law of environment operation.

3.3 Data Processing to Facilitate Administration

Data acquired from different sources (i.e., mainly from the physical environment and the Internet) are analyzed, integrated, and visualized. The massive data acquired from multiple sources is delivered using real-time databases such as Firebase Realtime Database³. This can help such data almost instantly to make effective decisions. Data analytics is essential to convert data into more factual information directly inquired by developers for decision-making. Furthermore, since environment data are collected from diverse sources, data integration helps discover the implicit patterns that cannot be uncovered

² Blender - [link](#).

³ Firebase Realtime Database – [link](#).



Fig. 2. Homogenous construction of mina tents [10]

based on a single data source. Moreover, data visualization technologies are incorporated to present data more straightforwardly. Finally, advanced artificial intelligence techniques can be incorporated to enhance a Digital Twin's cognitive ability (e.g., reasoning, problem-solving, and knowledge representation), so that specific, relatively simple recommendations can be made automatically.

3.4 Environment Behaviors Simulation in a Virtual Environment

The enabling technologies include simulation and virtual reality (VR). The former is used to simulate critical functions and behaviors of the physical environment in the virtual world. In the past, simulation technologies are widely used in environmental design. On the other hand, virtual reality (VR) technologies involve designers and even users to ‘directly’ interact with the virtual environment in the simulated environment. Recently, VR technologies are increasingly employed to support virtual prototyping and environment design [11]. Many readily available VR hardware devices can be directly adopted for digital twins. For this framework, HTC’s VIVE™ VR⁴ devices were used to provide users with an immersive experience. This methodology can be further enhanced by simulating virtual crowd in these 3D models to provide more information on the effectiveness of the 3D model [12].

3.5 Recommendation and Actions to the Physical World

Based on the digital twin recommendations, the physical environment is equipped with a capability, using various actuators, to adaptively adjust its function, behavior, and structure in the physical world. Sensors and actuators are the two technological backbones of a digital twin. The former plays a role in sensing the external world, whereas the latter plays a role in executing the desirable adjustments requested by the digital twin.

⁴ HTC® VIVE™ Cosmos Elite Series - [link](#).

In practice, the commonly used actuators that are suitable for consumer environments include, for example, hydraulic, pneumatic, electric, and mechanical actuators. Augmented reality (AR) technologies can also reflect some parts of the virtual environment to the physical world. For example, AR enables end-users to view the real-time state of their environments. Recently, AR technologies are increasingly applied in the factory domain production engineering [13].

3.6 Establishing Real-Time Connections Between the Physical and Virtual Environments

The connections are enabled using several technologies, such as network communication, cloud computing, and network security. Firstly, networking technologies enable the environment to send its ongoing data to the ‘cloud’ to power the virtual environment. The feasible networking technologies for consumers include, for example, Bluetooth, QR code, barcode, Wi-Fi, Z-Wave, and other similar technologies. Secondly, cloud computing enables the virtual environment to be developed, deployed, and maintained entirely in the ‘cloud’ conveniently accessed by both designers and users from anywhere with Internet access. Lastly, since environment data are directly and indirectly concerning user-environment interactions, it is critical to guarantee connections. Much effort has been devoted to connecting the physical and virtual environments in light of the Internet of Things, adapted for the digital twin research.

3.7 Multi-source Data Collection of Vital Information

Generally speaking, three types of environment-related data should be processed by the digital twin. For ordinary environments, physical environment data is usually divided into environmental data, customer data, and interactive data. Environment data contains customer comments, viewing, and download records. Interactive data consist of user-environment interaction, such as stress, vibration, etc. Using the sensor technology and IoT technology can gather some of the above data in real-time, web page customer browsing records, download records, evaluation response, can obtain the rest of the data. For this framework, one sensor used to collect data from the physical space is the Footfall® 3-D People Counter⁵. It provides vital information on the number of participants at a particular venue. The collected data are fed to Sect. 3.2 to close the loop towards building more functional virtual environments.

4 Implementation

One of the significant problems towards smarter cities is to develop multi-criterion decision support and prediction models for the authorities. These models can further be utilized to simulate multiple scenarios. Companies such as ESRI are actively developing applications that can assist in making decisions based on several criteria. One such tool is ArcGIS Urban that enables 3D urban environments for making significant decisions.

⁵ Footfall 3-D People Counters – [link](#).

Furthermore, ArcGIS City Engine assists with creating 3D content for Urban planning, making it more comfortable with making decisions and exporting the model in multiple 3D formats. Figure 3 demonstrates the usage of an exported 3D model of the Jamarat Bridge in ESRI ArcGIS Pro. This model can be further utilized in decision making application for better understanding of planned scenarios.



Fig. 3. Demonstrating digital twin at scale using ArcGIS Pro

Dashboards are also considered an essential component of a digital framework. It is extensively used to provide better insights for making an informed decision related to smart cities. The data for these dashboards are acquired for multiple sensors, active as well as passive in nature. Using these dashboards, experts can monitor the actual situations and changes in a location and make wiser decisions [14]. The data received from the sensors can be enormous that results in big data sets. These datasets need to be analyzed using a cloud-based GIS platform. The data can also be reduced by determining relevant and essential data and discarding the rest.

5 Discussion

As we move towards smart cities and intelligent cyber-physical systems, the resulting low-cost digital twin framework will provide an economically viable and suitable alternative to other options with incredibly high costs. The proposed framework is exceptionally suitable for homogeneous urban locations where physical structures are similar to one another. This scenario expedites creating a smart ecosystem of smart sensors and IoT devices with prediction systems to provide decision-makers with all the necessary information and make well-informed decisions.

There are certain limitations to this methodology. Initially, the point cloud scan quality depends on the device being used to collect details of the structures. Each Lidar device has limitations as to how many points it can collect in a minute. Expensive

devices have a more significant capability of collecting more points, but these devices incur colossal costs on the project. In comparison, low-cost devices with lower points per minute capability can provide enough details to generate a 3d model.

6 Conclusion and Future Work

In this paper, the authors presented a novel digital twin framework that is both economical and sustainable. Overall production and storage of spatial data demand innovation that can cater to a broader audience. Thus, it is necessary to provide real-time and precise information, including its spatial location. Such information flow is entirely automated and continuous, where the data should flow from the physical space to the virtual environment and communicate commands and recommendations back to the actual world.

The framework also intends to focus on significant data convergence of multi-sensor data and their connectivity with the virtual environment. The framework will also be tested for its accuracy in the 3D model's spatial details by collecting and comparing results from multiple low-cost lidar scanning devices. The primary outcome of improvements in real-time sensor information is the prediction accuracy. Simulating prediction models on such a digital twin platform can be used to plot current and forecast imminent hazards. These patterns can be used for accurate trend analyses. The development of a cost-effective digital twin of the urban area will guide present, and upcoming potential solutions against the administrative challenges through the steps discussed earlier.

Acknowledgment. The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia, for funding this research work through project number 0909.

References

1. Frost, L.A., Sullivan, D.L.: Smart cities. *Frost Sullivan Value Propos.* **1**(1), 42–49 (2019)
2. Wang, Y., Chen, Q., Zhu, Q., Liu, L., Li, C., Zheng, D.: A survey of mobile laser scanning applications and key techniques over urban areas. *Remote Sens.* **11**(13), 1540 (2019). <https://doi.org/10.3390/rs11131540>
3. Lee, K.W., Son, H.W., Kim, D.I.: Drone remote sensing photogrammetry. *Seoul Goomib* (2016)
4. No, H.-S., Baek, T.-K.: Real-time underground facility map production using Drones. *J. Korean Assoc. Geogr. Inf. Stud.* **20**(4), 2016–2017 (2017). <https://doi.org/10.11108/kagis.2017.20.4.039>
5. Guisado-Pintado, E., Jackson, D.W.T., Rogers, D.: 3D mapping efficacy of a drone and terrestrial laser scanner over a temperate beach-dune zone. *Geomorphology* **328**, 157–172 (2019). <https://doi.org/10.1016/j.geomorph.2018.12.013>
6. Krasikov, I., Kulemin, A.N.: Analysis of digital twin definition and its difference from simulation modelling in practical application. *KnE Eng.* 105–109 (2020). <https://doi.org/10.18502/keg.v5i3.6766>
7. Tao, F., et al.: Digital twin-driven product design framework. *Int. J. Prod. Res.* **57**(12), 3935–3953 (2019). <https://doi.org/10.1080/00207543.2018.1443229>

8. Sofia, H., Anas, E., Faiz, O.: Mobile mapping, machine learning and digital twin for road infrastructure monitoring and maintenance: Case study of mohammed VI bridge in Morocco. In: Proceedings - 2020 IEEE International Conference of Moroccan Geomatics, MORGEO 2020 (2020). <https://doi.org/10.1109/Morgeo49228.2020.9121882>
9. Tao, F., Cheng, J., Qi, Q., Zhang, M., Zhang, H., Sui, F.: Digital twin-driven product design, manufacturing and service with big data. *Int. J. Adv. Manufact. Technol.* **94**(9–12), 3563–3576 (2017). <https://doi.org/10.1007/s00170-017-0233-1>
10. Mina. – SoFlo Muslims. <https://soflosmuslims.com/mina/>. Accessed 01 Oct 2020
11. Stark, R., Israel, J.H., Wöhler, T.: Towards hybrid modelling environments—Merging desktop-CAD and virtual reality-technologies. *CIRP Ann.* **59**(1), 179–182 (2010). <https://doi.org/10.1016/j.cirp.2010.03.102>
12. Majid, A.R.M.A., Hamid, N.A.W.A., Rahiman, A.R., Zafar, B.: GPU-based optimization of pilgrim simulation for Hajj and Umrah rituals. *Pertanika J. Sci. Technol.* **26**(3), 1019–1038 (2018)
13. Nee, A.Y.C., Ong, S.K.: Virtual and augmented reality applications in manufacturing. *IFAC Proc.* **46**(9), 15–26 (2013). <https://doi.org/10.3182/20130619-3-RU-3018.00637>
14. Shirowzhan, S., Tan, W., Sepasgozar, S.M.E.: Digital twin and CyberGIS for improving connectivity and measuring the impact of infrastructure construction planning in smart cities. *ISPRS Int. J. Geo-Inf.* **9**(4), 240 (2020). <https://doi.org/10.3390/ijgi9040240>



VR in Heritage Documentation: Using Microsimulation Modelling

Wael A. Abdelhameed^(✉)

Applied Science University, Building 166, Road 23, Block 623, P.O. Box 5055, East Al-Ekir,
Kingdom of Bahrain
wael.abdelhameed@asu.edu.bh, wael.abdelhameed@fulbrightmail.org

Abstract. There are different digital methods and technologies that offer various applications and techniques in the field of heritage documentation and archiving. Virtual Reality (VR) is one of those technologies. VR has not been intensively used in heritage documentation, although its functions provide more potentials than other digital methods and technologies. The study through literature review classifies and reports different digital methods and technologies used in architectural heritage documentation. The study proceeds to present details of a VR function developed by the author. The newly developed VR function is utilized in heritage documentation of a case study. The case study is an existing archaeological site and its monuments, including their digital heritage resources of different formats, such as texts and images. The VR function enables to control the display of different narrations and chronological changes of the case study or part of it, in one VR model. Moreover, the VR function enables the user to visualise a certain era or a certain model-location, with display of relevant digital heritage resources -whether text or image-. The study concludes by proving the effectiveness of the VR function.

Keywords: Virtual reality · Microsimulation modelling · Architectural heritage · Heritage documentation · Microsimulation function

1 Introduction

The digital use in heritage documentation has become inevitable due to many advantages provided. In the field of architectural heritage there is no global standardisation of the heritage documentation, although several national institutions that deal with the documentation on the national levels developed a few data standards.

The main advantage of the 3D digital reconstruction of archaeological sites and historical buildings is the potential to document and archive the monuments in a more precise, efficient way.

2 Research Scope

The study focuses upon creating a VR tool for architectural heritage documentation, the study introduces a new VR function inside the microsimulation method (microsimulation player) of a commercial VR programme.

The study utilizes a case study, and develops its VR model including an archaeological site model, models of changes occurred in the site terrain, and a few monument models divided into different parts according to different eras, as well as heritage resources of different formats.

The VR function enables to chronologically display the substantial stages of excavation and reconstruction of the site's terrain and monuments, as well as simultaneously link the VR display to the heritage resources -texts or images- according to their relation of time and location.

3 Research Objectives

The research has the following two objectives:

- Developing a new VR function that controls the VR visualisation of the model's parts -both monuments and site-, as well as the VR visualisation of some heritage resources.
- Applying the new VR function in the case study's VR model, to prove its effectiveness in fulfilling requirements of the heritage documentation process, i.e. linking the VR display of chronological changes, to their heritage resources.

4 Literature Review

The architectural heritage documentation aims to enable the users to access and visualise the heritage information. Its processes are varied based on the monument's status: accessibility, size, complexity and existence. Beside the foregoing variables of the architectural monuments, there are other variables resulted from the available resources, which accordingly lead to select the modelling method and the used technology.

The digital reconstruction method is the process of digitally generating the 3D model of an object, by using the object's appearance, depth and colour [1]. Due to the large amount of data in a single project file, Boeykens et al. [2] had to divide the reconstruction process of architectural heritage into two stages, structure and detailing; the latter was also divided further into different files.

Methods of parametric modelling are used in historical reconstruction [3]. A procedural modelling method has been used in urban modelling by Parish and Müller [4], and in virtual heritage modelling by Müller et al. [5].

El-Hakim et al. [6] introduced a method to reconstruct and model large-scale heritage sites through which several technologies are employed: image-based modelling for basic forms and structures, laser scanning for fine details and surfaces, and image-based rendering for landscapes and surroundings.

VR technology is employed into a wide variety of applications and visual experiences. VR technology has fields, namely: augmented reality (AR), virtual reality, and mixed reality that refers to a combination of real and virtual elements the user can interact with each or even can virtually exists inside the model.

Sylaiou et al. [7] presented a system that integrates augmented reality with an interface tool and software to create web-based virtual museum exhibitions.

Number of techniques are associated with the AR systems such as photogrammetric, range-based modelling, image-based rendering and position orientation tracking [8].

From the literature review, various methods can be identified, some of which can be combined in one heritage documentation method based on the requirements and resources. In addition, one disadvantage appears that is the large file size, which leads to divide the documentation file into separate files.

5 Details of the Case Study

The research paper employs a case study to prove the new VR function effectiveness. The case study is a 5000-year-old site that includes two main forts and different structures, some of which had built over the ruins of older parts structures.

Although many archaeological ruins are identified with different functions: residential, public, and commercial, this research work and its VR model focus upon the main architectural monuments:

- The main architectural monument, on the top of the 12-m-height mound,
- the coastal fortress, and
- different terrains of different heights, representing a few excavation stages.

6 Advantages of the VR Function

Using the new VR function leads to have one VR model/platform with the following advantages:

- The VR platform includes different model-parts inside the same file, such as different terrains of the site and different models of each monument representing the changes occurred through time periods. Each model part of the main VR model reflects a time period and concurrently solves the issue of the large model size.
- The VR model acts as an architectural heritage documentation tool. Different narrations of the archaeological site can be displayed, as well as of the monuments, for example excavation of the terrain and reconstruction of the different monuments.
- The VR model documents and displays different heritage resources formats, such as images and textual explanation, according to their relevant time and location in the VR model narration.

7 VR in Architectural Heritage Documentation

7.1 VR Modelling Process

To achieve the flexibility in the heritage documentation for displaying different narrations, different parts/sub-components of the site and its structures/monuments are modelled in separate files by the 3ds max programme, including the different terrains and the related historical changes of structures/monuments.

The files of digital models -parts/sub-components- are separately imported with their textures into the VR Programme to make one VR model file. Developing the VR model of the case study site and its monuments starts with making the different terrains through historical periods, and proceeds by adjusting each monument's parts according to their coordinates in the site. The monuments' changes are added as well to the VR model. The links between these models' parts and the related heritage resources are created and stored in the VR model file.

A new VR function, therefore, is needed in the microsimulation player of the VR programme to control displaying these model parts/sub-components based on their chronology and topography. Images of the main narration of the case study are shown in Fig. 1.

7.2 Details of the New VR Function

The microsimulation modelling inside the VR platform allows the execution of XML algorithms. The researcher developed the new function from an older version [9] by XML code to control visualisation of the models' sub-components.

The method used in XML code is to make three variables of each sub-component inside the model: time, position, and direction with angle. The VR function utilizes the sub-component ID as the same as of the VR model.

The VR function controls the display time of each sub-component that is usually set up according to the chronological order and can be easily changed in the XML code to apply the VR display of another narration. The display position of each sub-component is set up to the different spatial parameters: X, Y, and Z inside the site, as well as the camera position inside the programme, Fig. 2. The time-control window inside the microsimulation player of the VR programme enables to pause the visualisation process which allows the user to navigate any VR model's part at any time point, Fig. 3. The navigation can be through walkthrough, flight mood or programme-camera control.

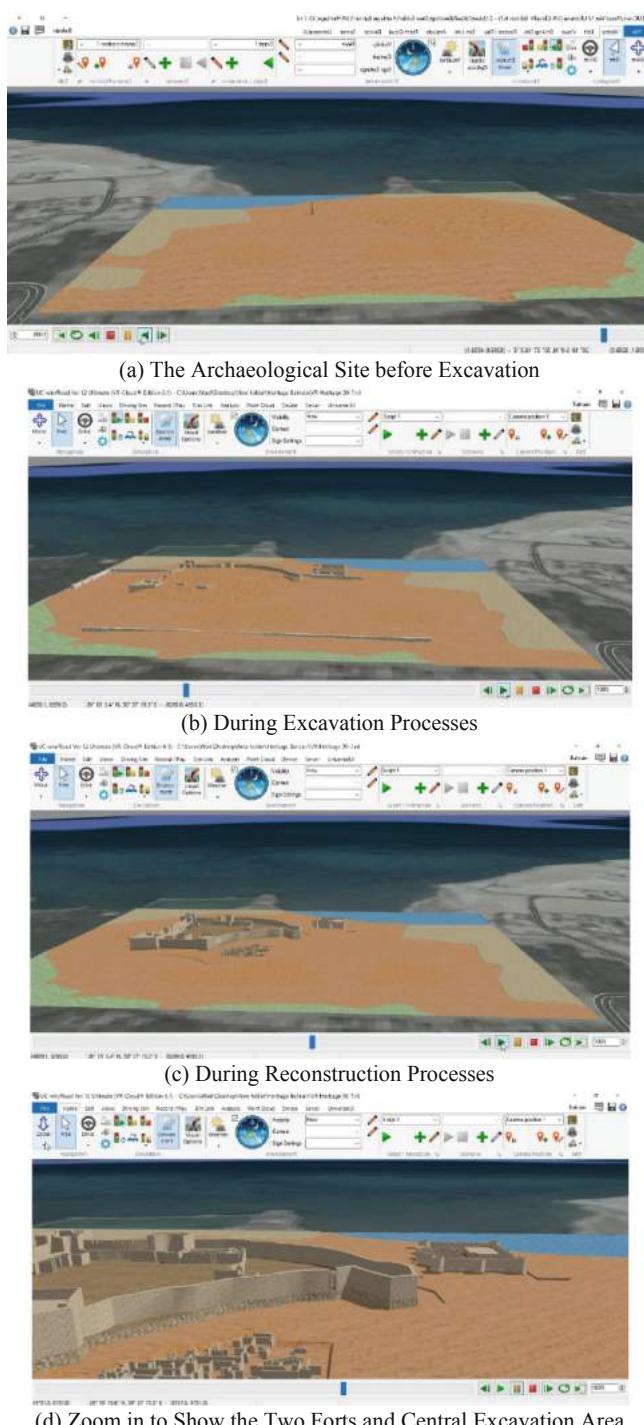


Fig. 1. VR Model of the archaeological site showing its main structures, screenshots of the microsimulation player window of the VR programme.



(a) The Main Fort During Reconstruction Processes from a Certain Viewpoint

(b) The Main Fort after Reconstruction Processes from the Same Viewpoint

Fig. 2. The VR model chronologically showing the reconstruction process of the main fort of the case study, screenshots of the microsimulation player window of the VR programme.

7.3 Microsimulation Modelling

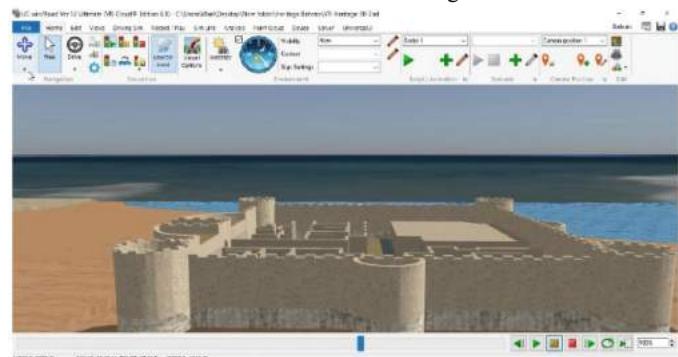
Using microsimulation modelling by dividing the monuments' models into sub-components enables to document and display the sub-components and their related heritage resources. The control of the sub-components inside the VR model and microsimulation player is achieved by XML algorithms. The algorithms of the new VR function control the display time of the heritage resources inside the VR model, in the relation to the programme-camera position and the microsimulation player time.

Using the VR function and microsimulation modelling overcomes the disadvantage of the large model that appeared in the literature review.

The effectiveness of the new VR function manifests itself in the control of different monuments and parts of one VR model, as well as the link of the VR display between the heritage resources and their relation of time period and in-model location, Fig. 4. The VR function is adapted and included in the commercial version of the VR Programme.



(a) Inside View of the Main Fort in the Walkthrough Mode of the VR Platform



(b) Outside View of the Coastal Fort in the Flight Mode of the VR Platform

Fig. 3. Walkthrough and flight modes in the VR model to display the model details, screenshots of the microsimulation player window of the VR programme.

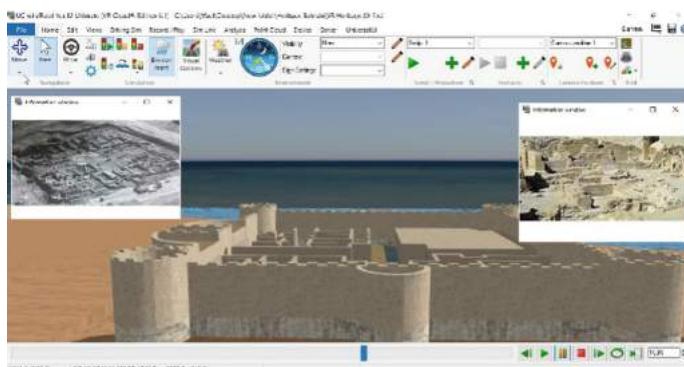


Fig. 4. Heritage resources' photos during the reconstruction process of the coastal fort, screenshots of the microsimulation player window of the VR programme.

8 Conclusion

The advantages behind using microsimulation modelling in the architectural heritage documentation is highlighted. The study proves the effectiveness of the new VR function.

The presented VR function can be employed into different disciplines and areas other than architectural heritage documentation, such as,

- developing an educational and a learning tool of the architectural history or even of the history in general, offers a more attractive method for learners and educators than the conventional methods. An interaction of the users with the proposed tool for education and learning can be developed, in order to fulfil all different learning styles: visual, aural, read/write and kinesthetic.
- making AR tours in the archaeological sites and the monuments. The user can physically be in the site, or away through a virtual model and the Internet. The physical presence of the user combined with the virtual presence in different eras, inside the architectural heritage model introduces a mixed use of VR and AR.

Acknowledgments. I would like to express my appreciation to the Forum8 Company and the software developers for the assistance they offered, in updating the programme Micor-simulation Plug-in to include my VR function.

References

1. Gomes, L., Bellon, O., Silva, L.: 3D reconstruction methods for digital preservation of cultural heritage: a survey. *Pattern Recogn. Lett.* (50), 3–14. Elsevier (2014). <https://www.sciencedirect.com/science/article/pii/S0167865514001032>. Accessed 6 March 2020
2. Boeykens, S., Himpe, C., Martens, B.: A case study of using BIM in historical reconstruction, the Vinohrady synagogue in Prague. In: Digital Physicality, Physical Digitality, eCAADe and CVUT, Faculty of Architecture, pp. 729–738 (2012). <https://core.ac.uk/download/pdf/34531367.pdf>. Accessed 6 March 2020
3. Chevrier, C., Perrin, J.: Generation of architectural parametric components: cultural heritage 3D modeling. In: Tidafi, T., Dorta, T. (eds) Joining Languages, Cultures and Visions: CAADFutures pp. 105–118 (2009)
4. Parish, Y., Müller, P.: Procedural modeling of cities. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH, ACM Press, Los Angeles, California, USA, pp. 301–308 (2001)
5. Müller, P., Vereenooghe, T., Wonka, P., Paap, I., Van Gool, L.: Procedural 3D reconstruction of puuc buildings in Xkipché. In: Proceedings of the 7th international Symposium on Virtual Reality, Archaeology and Intelligent Cultural Heritage - VAST2006, Nicosia, Cyprus, pp. 139–146 (2006)
6. El-Hakim, S., Beraldin, J., Picard, M., Godin, G.: Detailed 3D reconstruction of large-scale heritage sites with integrated techniques. *IEEE Comput. Graphics Appl.* **24**(3), 21–29 (2004). <https://ieeexplore.ieee.org/abstract/document/1318815/>. Accessed 6 March 2020
7. Sylaiou, S., Mania, K., Karoulis, A., White, M.: Exploring the relationship between presence and enjoyment in a virtual museum. *Int. J. Hum.-Comput. Stud.* **68**(5), 243–253. (2010). Elsevier, <https://www.sciencedirect.com/science/article/pii/S1071581909001761>. Accessed 3 June 2020

8. Noh, Z., Shahriza, M., Sunar, M., Pan, Z.: A review on augmented reality for virtual heritage system, 50–61 (2009). Springer, Berlin, Heidelberg. https://link.springer.com/chapter/https://doi.org/10.1007/978-3-642-03364-3_7. Accessed 3 June 2020
9. Abdelhameed, W.A.: Micro-simulation function to display textual data in virtual reality. *Int. J. Archit. Comput.* **10**(2), 205–218 (2012)



MQTT Based Power Consumption Monitoring with Usage Pattern Visualization Using Uniform Manifold Approximation and Projection for Smart Buildings

Ray Mart M. Montesclaros^(✉), John Emmanuel B. Cruz, Raymark C. Parocha,
and Erees Queen B. Macabebe

Department of Electronics, Computer, and Communications Engineering, School of Science
and Engineering, Ateneo de Manila University, 1108 Quezon City, Philippines
`ray.montesclaros@obf.ateneo.edu, emacabebe@ateneo.edu`

Abstract. The persistent electricity price hikes in the Philippines and the impact of climate change on the country's energy demands adversely affect the consumer's finances on a regular basis. These form the foundation for energy efficiency initiatives in building-based electricity consumption. Such initiatives encompass a wide range of innovations from energy generation to real time monitoring, management and control. At its core, this study supplements institutional efforts on energy management through the electricity consumption monitoring system. A non-intrusive data acquisition system that monitors the aggregate electricity consumption and visualizes the appliance usage patterns in a building setting was developed. This Non-Intrusive Load Monitoring (NILM) technique allowed acquisition of data from a single point of measurement using only a single sensor clamped to the main powerline. The acquired data were streamlined to communicate with the IoT OpenHAB framework via Message Queuing Telemetry Transport (MQTT) protocol and were implemented through deployment. Lastly, Uniform Manifold Approximation and Projection (UMAP) was used for dimension reduction. UMAP was applied to the raw time series data of the aggregate power consumption in order to visualize and determine appliance usage patterns, and effectively label data instances through a scatter plot.

Keywords: Energy monitoring · NILM · IoT · OpenHAB · Smart buildings · UMAP · Dimension reduction

1 Introduction

The emergence of “Smart Living” paved the way for people to use the available technologies to enhance their living experience. Two key concepts are closely related in Smart Living; Internet of Things (IoT) and Machine Learning. Incorporating the two allows improvements in the quantity and quality of data gathered and turns this data into useful and actionable information. The concept of smart living gave way to other smart

concepts, one of which are smart buildings. A smart building is any structure that makes efficient use of automated processes in order to optimize and minimize energy usage [1]. To make these optimizations possible, accurate and reliable data are crucial and can be obtained through monitoring systems while also considering cost-efficient setups.

The Philippine energy consumption has been on the continuous rise in the past few years. Data show that there has been an increase by 4% to 6% in electricity consumption compared to the previous year [2]. Electric meters are usually positioned at locations that are hard to reach or concealed which makes regular manual monitoring inconvenient. This prevents consumers from obtaining information and incapable of managing their energy usage efficiently. In order to address these issues, load monitoring systems have been developed so that consumers can obtain their power data and have a better understanding of their electricity consumption behavior [3]. The system utilized the Non-Intrusive Load Monitoring (NILM), a low-cost, non-intrusive energy measuring system which records the aggregate power consumption of a residence or a building without interfering with the existing electrical system.

This study presents the design and development of a data acquisition system for electricity consumption using NILM. This system was integrated to the MQTT protocol and the acquired data were streamlined to communicate with the OpenHAB framework for ease of access. The Uniform Manifold Approximation and Projection (UMAP) was employed for dimension reduction in order to visualize and determine appliance usage patterns and effectively label data instances through a scatter plot based on the raw time series data of the aggregate power consumption.

2 Theoretical Background

2.1 Total Aggregated Power Consumption

The total aggregated power consumption states that the individual power consumption of each appliance can be obtained by:

$$p_t(t) = \sum_{i=1}^n p_i(t) + e(t) \quad (1)$$

Where $p_t(t)$ is the aggregated power consumption, $p_i(t)$ is the energy consumption of an appliance, and $e(t)$ is the measurement errors and line loss [2]. The aggregated power consumption indicates that the total average power consumption is just the summation of the power consumption by each individual appliance. Figure 1 shows power consumption patterns of individual appliances and combinations of these appliances. Each appliance displays a unique characteristic called “digital signature”.

As shown in Fig. 1, some appliances consume a lot more power compared to the other while some have different states with regards to power consumption. The digital signature of each appliance can be used to determine which specific appliance contributed to the aggregate power consumption [3]. These digital signatures can be attributed to the operational state of the appliance. Type-I; Single-State or Two-State appliances operate in two states. Type-II; Multi-State appliances or Finite-State Machines (FCM) have more than two operating states, where each state has different power consumption compared to the other states. Type-III; Continuously Varying State, where appliances under this state do not have a well-defined and no fixed number of states [4].

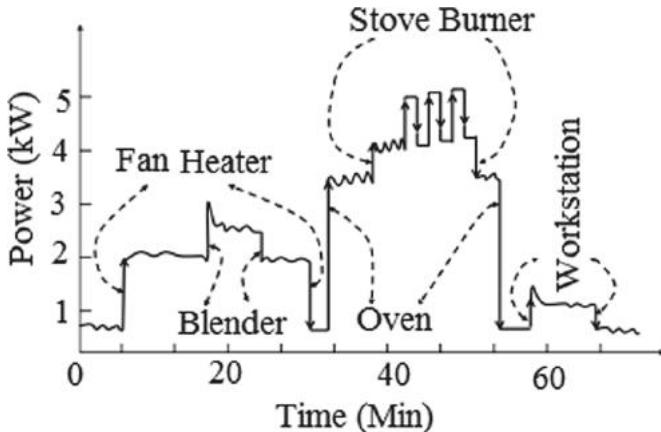


Fig. 1. Power vs time plot of the total power consumption [3].

2.2 MQTT

MQTT stands for Message Queuing Telemetry Transport, which is a machine-to-machine (M2M) /'Internet of Things' connectivity protocol and was designed as an extremely lightweight publish/subscribe message transport [5, 6]. MQTT works on the TCP layer and delivers messages from node to server and is ideally used for the (Internet of Things) IoT nodes which have limited capabilities and resources [7]. It consists of two message sets on a connection 'Publish' and 'Subscribe' [8]. Any data that is published under a certain topic is received by any entity subscribed to that topic.

2.3 Time Series Clustering and Dimensionality Reduction

A time series is a series of data points that is ordered or graphed with respect to time. It is a sequence of data points that is taken at equal intervals of time [9]. Time series clustering are methods where the long time series data sets were clustered so that the data characteristics and features can be extracted [10].

Dimensionality reduction is the process of reducing the dimensions of a time series to enhance the efficiency of extracting patterns in the data [11]. A time series C of length n can be reduced to w dimensions by:

$$\bar{C}_l = \frac{w}{n} \sum_{j=\frac{n}{w}(i-1)+1}^{\frac{n}{w}} C_j \quad (2)$$

where, \bar{C}_i is a piecewise aggregate approximation of time series vector. Using this formula, the data is divided into w equal sized frames. The mean value of the data that falls within the frame is calculated, and the vector is obtained. The vector of the calculated values becomes the data-reduced representation [12].

Uniform Manifold Approximation and Projection (UMAP) is a manifold learning technique developed for dimension reduction. In dealing with high dimensional data, UMAP creates a topological representation using local manifold approximations, and

then patches their local fuzzy simplicial set representations, which were constructed from two processes: (1) approximation of a manifold which the data is assumed to lie, and (2) construction of a fuzzy simplicial set representation of the said manifold. Given a low dimensional representation of the data, the same set of processes will be used to construct an equivalent topological representation. UMAP then will optimize the layout of the data representation in low dimensional space. This is to minimize the cross-entropy between the two representations [13].

3 Methodology

3.1 System Architecture

The power consumption monitoring system was developed to measure the electricity consumption in an electrical network. Figure 5 shows the Power Consumption Monitoring System Architecture. The system architecture consists of six main parts as shown in the figure. The locally connected devices in a computer laboratory within a school building where the system was deployed sends their aggregate current reading to the

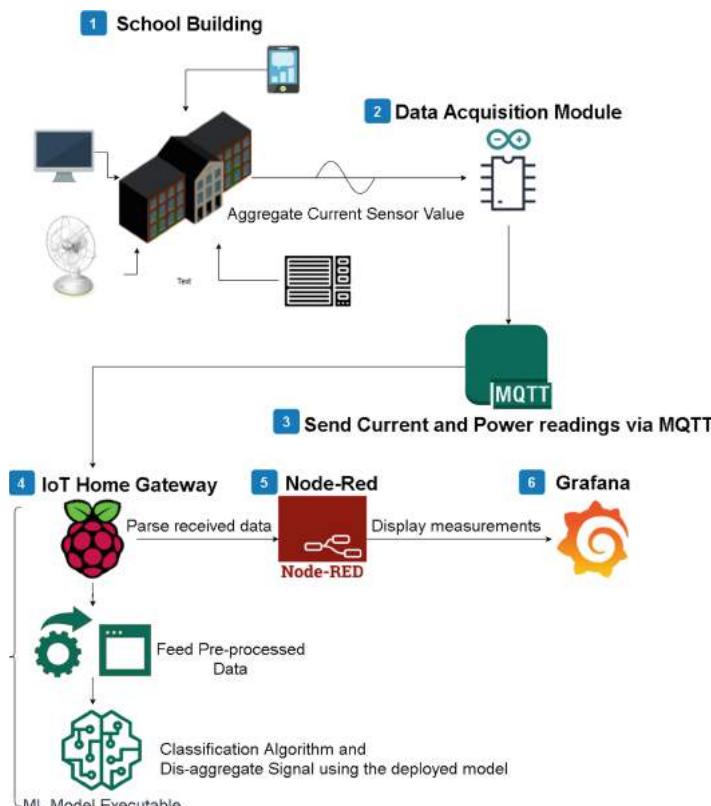


Fig. 2. Architecture of the power consumption monitoring system.

Data Acquisition Module via a non-invasive current sensor. Then, the module sends the data to the Raspberry Pi (RPi) which serves as the IoT Gateway. The data, which contain power readings, undergo preprocessing for classification using dimension reduction via UMAP. From the IoT Gateway, the total power consumption readings and the visualized usage patterns are sent and displayed in a Grafana instance using nodeRed via MQTT. The dashboard then displays the individual power consumption readings on each sensor and the total power consumption in kWh.

3.2 Hardware System Setup

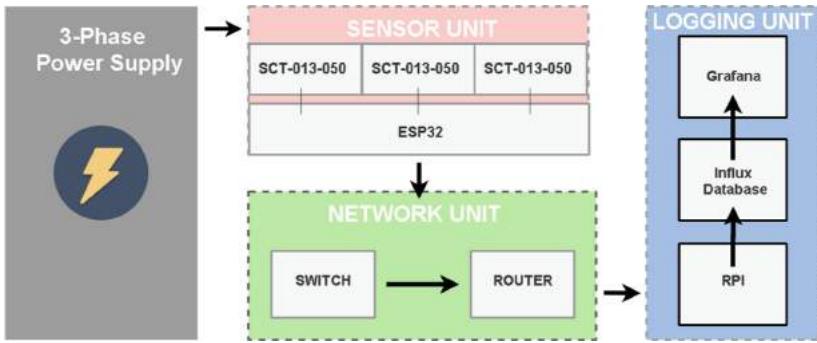


Fig. 3. Hardware system block diagram.

The block diagram for the system's hardware setup is shown in Fig. 3 and is based on the Semi-Automated Data Acquisition (SADA) system used by Bugnot and Macabebe [14] with a few modifications. The socket array unit and Arduino microcontroller were removed, and the data acquisition module was designed to read current values from a 3-phase power supply. The module allows non-intrusive gathering of current readings and calculates the aggregate power consumption for pre-processing and time-series clustering.

The aforementioned values were sent through a localized network with static IP connection. Since Node-Red runs locally and subscribes to the same topic as the one being published by the data acquisition module, the logging unit stores the received and parsed data in InfluxDB. Lastly, the data stored is then displayed in a Grafana dashboard. The setup is divided into three main blocks based on their corresponding functionalities, namely: the sensor unit, the network unit and the logging unit.

The sensor unit is composed of three SCT-013-050 non-invasive split-core current sensors and an ESP32 microcontroller. This unit was used to gather aggregate current data from the circuit breaker. The sensor has a rated input range of 0 A to 50 A rms, but can handle up to 60 A rms, and an output of 1 V AC rms. The sensor has a full scale accuracy of $\pm 1\%$ and operating frequencies of 50 to 1000 Hz. The aggregate power consumption was obtained by assuming 220.63 VAC as the voltage multiplier to the aggregate current. This voltage multiplier was the measured reading from the digital multimeter.

The network unit is mainly composed of a switch and a router. This unit was intended for the purpose of transferring data packets between multiple network devices. This setup allows for easy access when troubleshooting connectivity issues and connects to the RPi remotely via ssh.

Once the data from the ESP32 was successfully received by the RPi, the Node-Red instance would then parse the received character array data type to its corresponding double-precision value, and then store the parsed data to the database. To display the necessary metrics, Grafana was used. It loads the data that was stored in the database in InfluxDB, and outputs a real-time visualization for the current reading of each sensor, and the aggregate power consumption. Figure 4 shows the user interface data visualization using Grafana.



Fig. 4. Real time data visualization using Grafana.

3.3 Software System Flow

The flow diagram for the software system is shown in Fig. 5. The data acquisition portion sends the raw time series data for pre-processing. The pre-processing portion converts the raw data to a form that can be inputted to the UMAP object.

Once the preprocessed data is passed as an argument, the reducer is then trained to learn about the manifold. UMAP utilizes sklearn's API in order to use the fit method and indicate which dataset the model needs to learn from [15]. The data points are then embedded on a 2D-plane. The usage patterns of the appliances are then visualized using a standard scatterplot. Visualization was done by clustering reading with similar patterns using the processed raw time series data. The pre-processing methodology for dimension reduction on temporal data for visual analytics was used. The first part performs unity-based normalization. This ultimately sets the raw data to a common scale without distorting the differences in the values [11].

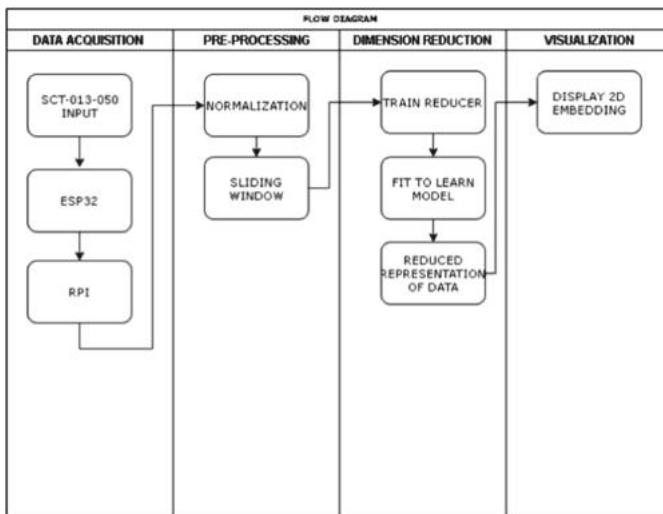


Fig. 5. Software flow diagram.

After the values were normalized, the Sliding Window Technique was implemented. This technique was done by incrementing the bounds of the observation window in the time-domain and as a result, also sliding the observation window across the length of the collected data-sample. This converts a 1D stream of data with N columns into a $(N-(W-1)) \times W$ matrix where W is the window size. It was observed that as the window size increases the clusters become more indistinguishable. This can be attributed to the fact that as the size of the observable window increases, more unique functions are taken into consideration making it more difficult to cluster data points with similar patterns. Since the window size was up to the proponents' discretion, four window sizes were analyzed (2, 4, 60, 300). Among these sizes, a window size of 4 produced the most distinguishable clustering with the given dataset.

In this case, the researchers used a window size of 4 which produced an $(N - 3) \times 4$ matrix.

After this, the resultant matrix from the sliding window technique underwent dimension reduction. The purpose was to provide additional visual analysis and to reduce the feature space into a 2D plane without compromising the shape characteristics of the original data. Through dimension reduction, the structure of the original data is preserved and it allows us to draw out certain intuitions about the data itself through visual means. This task can easily be done by using UMAP. After instantiating the UMAP class, the reducer was then trained to learn about the manifold. The fit method accepts the resultant matrix in order for the model to learn from it. The reducer accepts certain parameters before embedding the data into a 2D plane. After the reducer was trained, the transformed data was visualized. This was done by the transform method available in UMAP. After which, a UMAP projection was displayed by plotting the resulting embedding.

4 Results and Discussion

4.1 Data Collection Results

The data acquisition module shown in Fig. 2 was deployed in a computer laboratory within the university. The computer laboratory had the following devices: 100 AT and 100 AF circuit breaker, (1) switch, (1) router, (1) projector, (2) air-conditioning units, (4) lights, and (24) desktop computers. The data acquisition module requires a power supply with 5 V input voltage and a recommended input current of 2 A. The module operates inside the circuit breaker. This was done by tapping into one of the breaker bus bars. The computer laboratory layout showing the devices in the room is seen in Fig. 6.

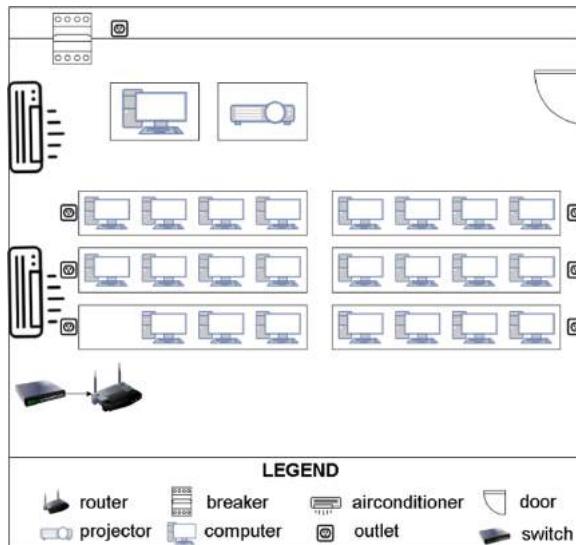


Fig. 6. Computer laboratory room layout.

Current Sensor Readings. There were three current sensors that were deployed and labeled for each corresponding phase wire inside the power box. Figure 7 shows the location of the current sensors connected to the wires in the circuit breaker box. Looking at the figure and considering the current reading of the three phase wires, the right and middle phase wires follow the same trend of consumption, which can be attributed to the fact that the breaker switches for the appliances were supplied by the said phase wires, making it contribute the most in the total power consumption. Each phase wire can supply 110 V, making two phase wires necessary to supply 220 V for each switch. The left phase wire shows a way less current consumption, and has a near constant current reading; different from the trend of the other two phase wires.

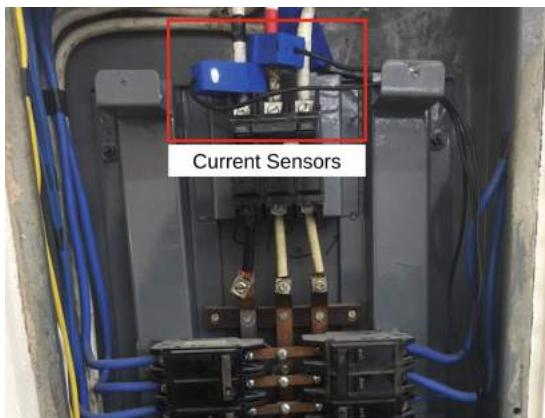


Fig. 7. Computer laboratory room layout.

Computer Laboratory Power Consumption Readings. The power consumption data was gathered. Figure 8 shows a sample power consumption data of the room within a day; where trends can be observed. A general observation from the data gathered shows that there was a spike in power consumption at the start of the day, at around 8:00 am, where all appliances were powered on. This huge spike was mainly caused by the air-conditioning units which take time to stabilize given the initial temperature within the room.

Moreover, the power consumption of the room is relatively stable, but it can be observed that there is a rising and falling trend. It can be attributed to the compressor of the air conditioner which cycles on and off to complete a cooling cycle.

Lastly, the schedule of the room greatly affects the power consumption, as the power consumption rises when there is a class being held inside, since the appliances were being used.



Fig. 8. Computer laboratory power consumption.

Cost Computation. One of the objectives was to be able to design and develop a system that monitors building-based electricity consumption. In order to do this, it is imperative for the system to provide its users not only the necessary power consumption metrics but also its corresponding costs.

The cost computation is based on certain assumptions and estimations obtained from the summary scheduled rates published by the electric utility company [16]. The service rate is used by Meralco to classify its customers and to appropriate their corresponding bills based on the approved rate schedule [17]. The service rate classification used for the computer laboratory was General Service A (GS-A) under the 0–200 kWh criteria. Table 1 is a list of charges and their corresponding rates used to calculate the electricity rate per kWh.

Calculating the total energy consumption for the power consumption data shown in Fig. 8, the total energy consumed during this day was calculated at 59.078 kWh amounting to Php 480.51.

Table 1. Summary of charges and scheduled rates.

Service Rate Type		
General Service A (GS-A) (0–200 kWh)		
Charge Type		Rate (Php/kWh)
Generation Charge		4.6632
Transmission Charge		0.7859
Distribution Charge		1.0012
Supply Charge		0.5085
Metering Charge		0.3377
System Loss Charge		0.3528
Universal Charge	UC-ME	0.1561
	UC-EC	0.0025
	UC-SCC	0.1453
	UC-SD	0.0428
Fit-All (Renewable)		0.0495
Lifeline Rate Subsidy		0.0880
Senior Citizen Subsidy		0.000
TOTAL RATE		8.1335

Test Cases. There were eight random combinations that were considered as test cases. For each test case, data were gathered for 15 min and the total power consumption in each case was obtained. The purpose was to gather data labels that characterize usage patterns of the appliances. Table 2 shows the combinations of appliances present for each case. Note that there are four desktop computers in a row as seen in Fig. 6.

Table 2. List of test cases and the state of the appliances.

Case	Appliance			
	Aircon (No. and Temp.)	Lights (On/Off)	Computer Rows (No.)	Projector (On/Off)
1	2 at 16 °C	On	6	On
2	1 at 16 °C	On	6	On
3	0	On	6	On
4	2 at 23 °C	Off	4	On
5	2 at 23 °C	Off	0	Off
6	2 at 23 °C	On	0	On
7	1 at 16 °C	On	6	On
8	0	Off	4	Off

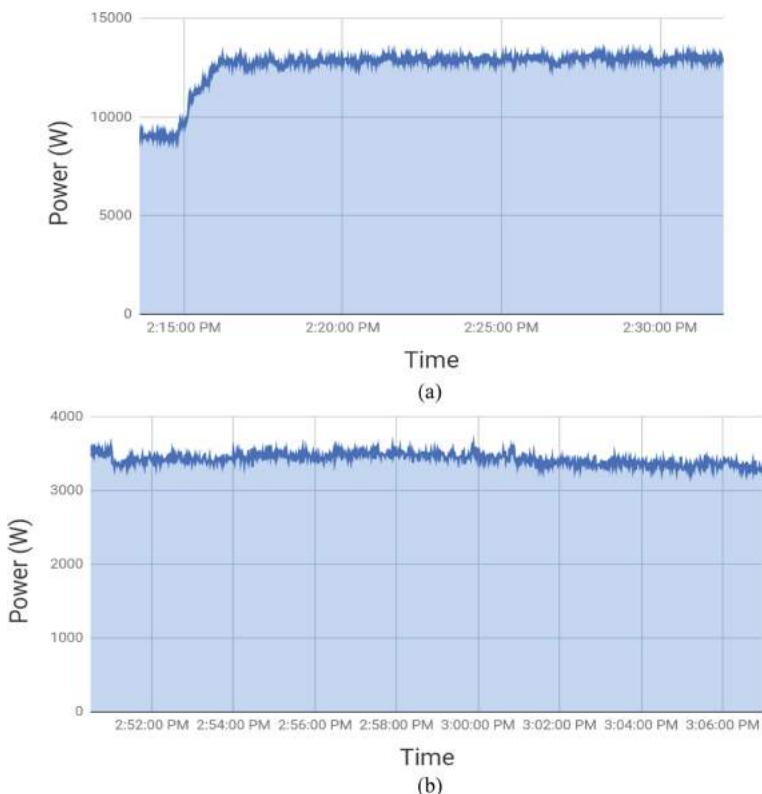
**Fig. 9.** Power consumption vs. time plots of (a) Case 1 and (b) Case 3.

Figure 9 shows the power consumption graph of (a) Case 1 and (b) Case 3. Comparing the graphs of Case 1 and 3, it can be seen that the major contributor to the power consumption of the room are the air conditioners. Looking at the difference in power consumption, it can be said that the contribution of the air conditioners alone totals to 9,000 W to 10,000 W.

4.2 Preprocessing Results

The raw time series data were preprocessed to convert them to a form that can be used as input to the UMAP object. In this section, the outputs of the unity-based normalization and sliding window technique are discussed and analyzed.

Normalization. The type of normalization used for the raw time series data was unity-based normalization. The purpose is to set all the values in the dataset within a range of [0, 1]. Figure 10 shows an example of a normalized power consumption data.

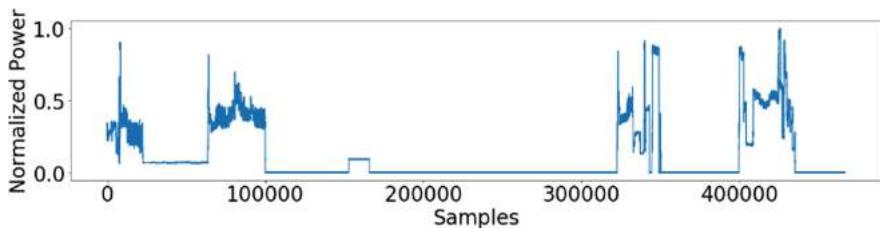


Fig. 10. Normalized power consumption readings.

Since there were 466,629 data points, a subset was also normalized to improve the speed of the analysis. In this case, the subset spanned data points from 12:00 PM until the last element of the training dataset. This was observed to be the time where the data acquisition module had the most stable readings. The whole dataset was split into two, 80% of the dataset was used for training and 20% of the dataset was used for testing.

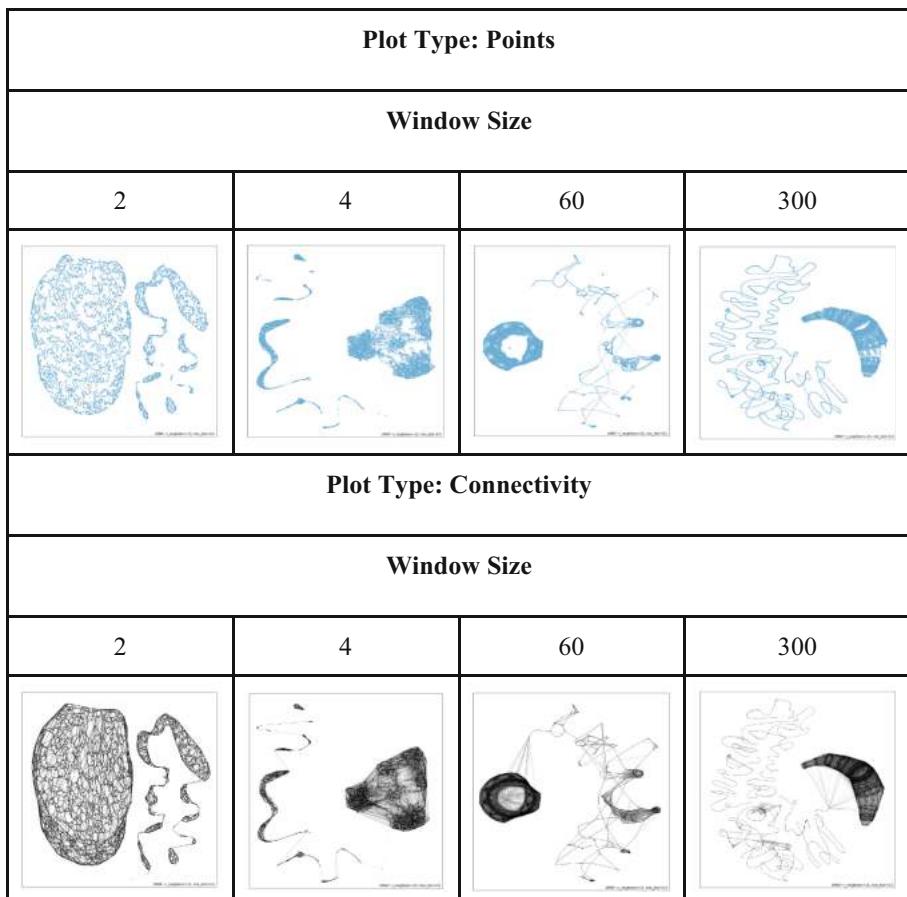
Sliding Window Technique. This process converts a 1D stream of data with N columns into a $(N - (W - 1)) \times W$ matrix where W is the window size. For this study, a window size of 4 was chosen and produced an $(N - 3) \times 4$ matrix. Increasing the window size to a higher value drastically increased processing time and as a result slowed down the embedding and visualization process.

The rationale behind using this technique was to extract more information about the behavior and the usage patterns of the appliances being used. More information can be extracted when the dataset is a continuous stream of points within a certain window size compared to independent or discrete data points within that same window size. The sliding window technique was able to preserve the original behavior of the dataset and can be used as input to the UMAP object.

4.3 Appliance Usage Patterns

After preprocessing, the data served as input to the UMAP object for visualization using the `umap.plot` package. First, the window size was varied in order to choose a plot that produced the most distinct clusters. For each window size, the following plots were used to visualize the trained UMAP model: (1) points, and (2) connectivity. The connectivity plots were used to visualize how the connectivity of the manifold relates to the resulting 2D embedding since UMAP is a topological representation of the data's manifold for which it was sampled from. All plots had a `random_state = 42`, a `min_dist = 0.0`, `n_neighbors = 15` and a `min_dist = 0.0`.

Table 3. Appliance usage pattern point and connectivity visualizations with increasing window sizes.



As seen in Table 3, the clusters become more indistinguishable as the window size increases. This can be attributed to the size of the observable window. As the size of

the observable window increases, more unique functions are taken into consideration making it more difficult to cluster data points with similar patterns. In this study, four window sizes were analyzed. Among these sizes, a window size of 4 produced the most distinguishable clustering with the given dataset A window size of 2 produced clusters that were in close proximity to each other.

Figure 11 shows the manifold projection of the power consumption dataset using window size = 4. In the figure, there are four main clusters formed and nine clusters in total (subclusters included). Since the data labels gathered were limited, partial labeling or semi-supervised learning using UMAP was used. Random masking was performed on target labels and the unlabeled data points were given a value of -1 , this is an sklearn standard for unlabeled target information. After this, supervised learning was applied to the dataset and it will learn accordingly by considering entries with -1 as unlabeled.

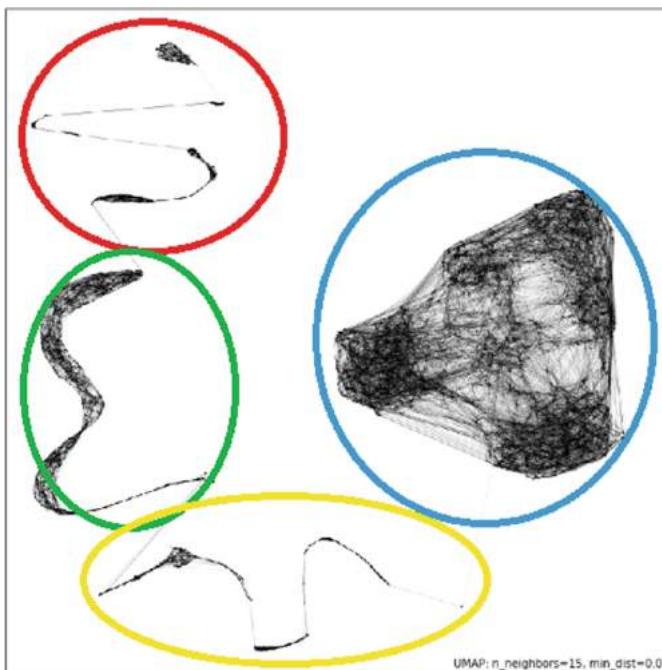


Fig. 11. Manifold projection of the power consumption dataset using window size = 4, showing 4 distinct clusters.

The embedding seen in Fig. 12 shows the resulting topographical representation when semi-supervised learning is applied to a smaller dataset. The eight data labels used were the appliances turned ON for a duration of 15 min per test case. As seen in the figure, similar data points were closer to each other. Although two ACs contributed highest in terms of power consumption, the cluster was still smaller compared to the “12 computers” case since most of the cases performed included computers being turned ON more frequently than cases which included ACs alone. However, when combining

all the clusters which include ACs, it is apparent that these appliances contribute to most of the power consumption readings inside the computer laboratory.

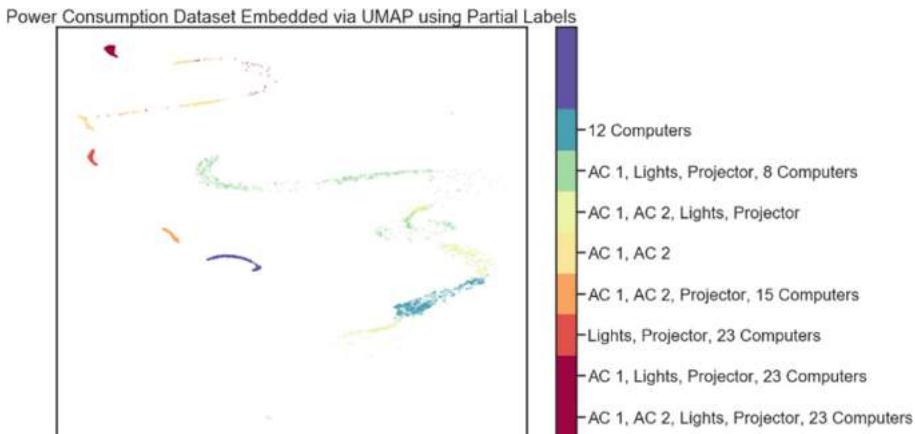


Fig. 12. Power consumption usage patterns embedded via UMAP using partial labels on a day's worth of data.

5 Conclusion

This study presented an MQTT based power consumption monitoring system that can measure and display the power consumption in real time. The data-acquisition module can measure the aggregate power consumption of a three-phase power supply. Moreover, it was successfully streamlined to the IoT home gateway over a protected local network via MQTT protocol. The appropriate current and power metrics were then displayed on a Grafana dashboard.

The power consumption data was analyzed to extract insights about the behavior of the appliances present in the computer laboratory. The energy consumption in kWh and the corresponding amount were determined using the service rates of the utility company. Using the data from the test cases, the two air conditioning units were identified as the major contributors in the increase and sudden fluctuations of the power readings.

Lastly, the raw time series analysis was conducted by applying unity-based normalization and sliding window technique to the data prior to sending it to the UMAP object for dimension reduction. This helped visualize appliance usage patterns. A variable window and step size provides flexibility which allow modifications in the dimension reduction process without compromising the original behavior of the dataset. Through the 2D topological representations, it was observed that data instances which exhibited similar patterns tend to form clusters. This can be used as a basis to associate repetitive patterns. On the other hand, observed outliers can be used to streamline the identification of anomalies in the appliances being used inside the vicinity.

As future work, categorical label information can be included to perform supervised dimension reduction. This will allow the system to classify appliances in real time

based on the device's usage pattern. Automating the process of exhausting all possible combinations of appliance activation will reduce the data-collection procedure. Further improvements on testing and selecting other features for appliance recognition can speed up training time, reduce the complexity of the model, and increase the model's accuracy.

Moreover, hardware refinement by limiting and simplifying the design of the data acquisition module can reduce costs.

Lastly, when scaling energy monitoring systems for buildings it is imperative to look into IoT fleet management platforms which will allow continuous development and updating of source code on multiple deployed devices. This will also allow admins to monitor multiple devices through a dashboard containing all the necessary metrics.

Acknowledgment. This study was funded by the University Research Council (URC) of the Ateneo de Manila University under the URC project, "Development of an Energy Monitoring System for Smart Buildings (Phase 1)."

References

1. What is a smart building and how it can benefit you?. <https://www.rcrwireless.com/20160725/business/smart-building-tag31-tag99#:~:text=A%20smart%20building%20is%20any,lighting%2C%20security%20and%20other%20systems>. Accessed 14 Jan 2021
2. Doe.gov.ph. https://www.doe.gov.ph/sites/default/files/pdf/energy_statistics/05_2018_power_statistics_as_of_29_march_2019_electricity_consumption.pdf. Accessed 05 Oct 2019
3. Aladesanmi, E., Folly, K.: Overview of non-intrusive load monitoring and identification techniques. IFAC-PapersOnLine **48**(30), 415–420 (2015)
4. Zoha, A., et al.: Non-intrusive load monitoring approaches for disaggregated energy sensing: a survey. Sensors **12**(12), 16838–16866 (2012)
5. FAQ - Frequently Asked Questions: MQTT. <http://mqtt.org/faq>. Accessed 08 Oct 2019
6. MQTT. <http://mqtt.org/>. Accessed 08 Oct 2019
7. Hivemq. <http://www.hivemq.com/blog/mqtt-essentials-part-1-introducing-mqtt>. Accessed 08 Oct 2019
8. Mqtt version 3.1.1 becomes an oasis standard. <http://www.oasis-open.org/news/announcements/mqtt-version-3-1-1-becomes-an-oasis-standard>. Accessed 08 Oct 2019
9. The Complete Guide to Time Series Analysis and Forecasting. <https://towardsdatascience.com/the-complete-guide-to-time-series-analysis-and-forecasting-70d476bfe775?gi=5fe5401e38cd>. Accessed 16 Apr 2020
10. Wang, X., et al.: Characteristic-based clustering for time series data. Data Min. Knowl. Disc. **13**(3), 335–337 (2006)
11. Ali, M., Jones, M.W., Xie, X., Williams, M.: TimeCluster: dimension reduction applied to temporal data for visual analytics. Vis. Comput. **35**(6–8), 1013–1026 (2019). <https://doi.org/10.1007/s00371-019-01673-y>
12. Lin, J., et al.: Finding motifs in time series. In: Proceedings of the Second Workshop on Temporal Data Mining, vol. 3 (2002)
13. McInnes, L., et al.: UMAP: uniform manifold approximation and projection. J. Open Source Softw. **3**(29), 1–4 (2018)
14. Bugnot, R.J., Macabebe, E.Q.B.: A non-intrusive appliance recognition system. In: IEEE International Conference on Internet of Things and Intelligence System (IoTaIS) (2019)
15. UMAP 0.4 Documentation. https://umap-learn.readthedocs.io/en/latest/basic_usage.html. Accessed 12 May 2020

16. Meralco. https://meralcomain.s3.ap-southeast-1.amazonaws.com/2020-03/03-2020_rate_schedule.pdf?null. Accessed 12 May 2020
17. Meralco Customer Community. <https://online.meralco.com.ph/customers/s/article/Service-Rate-Classifications>. Accessed 12 May 2020



Deepened Development of Industrial Structure Optimization and Industrial Integration of China's Digital Music Under 5G Network Technology

Li Eryong¹(✉) and Li Yukun²

¹ Jiangxi University of Finance and Economics, Nanchang 330000, China
87036401@qq.com

² Universiti Putra Malaysia, 43400 Kuala Lumpur, Malaysia

Abstract. While 4G technology is developing vigorously, 5G construction has already been in full swing. It has attracted the attention of the industry for its advantages of fast transmission rate, low latency and mass connection of the Internet of Things. The change of 5G will prompt all walks of life to actively deconstruct and restructure related industries. The deep integration of 3D, VR, AR, AI and 5G technology will bring great changes in the creation, performance, dissemination and appreciation of digital music. This paper analyses the industrial structure changes and future development trend of China's digital music industry under 5G network technology, aiming to provide some useful new ideas for China's digital music industry.

Keywords: 5G · Digital music industry · AI · Block Chain

1 Development Background and Status Quo of China's Digital Music Industry

Digital music refers to music that uses digital technology in the production, distribution, and storage of music. It is mainly transmitted, downloaded, or enjoyed through the network. A computer or a mobile digital audio player decodes and issues digital music content in the cloud or downloaded MP3, WAV, FLAC, MPEG-2, AIFF and other formats to complete the transmission and consumption of the music. The advantages and characteristics of its immediacy, reproducibility, mass downloading and unlimited play effectively expand the range of music propagation and increase the speed of it.

This article mainly reviews the development of China's digital music market in the era of online music and wireless music. Online music refers to digital music that can be viewed online, or that can be downloaded directly to a computer and transmitted to other playback devices. Wireless music refers primarily to music services distributed in the Internet, including mobile phone ringtone downloads, coloring ring back tones, music downloads and wireless audio-visual music services.

The era of online music (1999–2008): In the mid-1990s, the technology of MPEG Audio Layer 3 compressed music with a compression ratio of 1:10 or 1:12, easily compressing a CD into a few megabytes. Digital technology has not only changed the survival mode of various fields, but also quietly guided the transformation of the traditional music industry. It greatly reduces the burden of carrying traditional music products (vinyl records, tapes, CDs), and the form of music carriers has radically changed with the development of technology. However, at the end of the 20th century, China's bandwidth speed was basically maintained at about 100 kb, download speeds were maintained at tens of kb, and users with multimedia computers accounted for very few. Therefore, online music did not form a scale and influence that surpassed the mobile ring back tone and record market.

In the early 21st century, the market size of online music has gradually expanded, the reasons to which are the popularity of electronic computers, the increase in network speed, the emergence of mobile music players, and the establishment of P2P online music websites. Typical music websites such as Kugou and Baidu have average daily downloads of 5 million to 10 million in 2005. The new business model is mainly based on free download and trial listening, and paid music. However, a series of problems have always restricted the development of the digital music market: for example, the digitization of music has brought unprecedented piracy problems, and the user's awareness of paying has not been cultivated, which has restricted the development of digital music to a certain extent. Since 2008, China's digital music market has entered a regulatory adjustment period. At the same time, WAP and 3G websites have been increasing and the channels for to have wireless music have been continually expanded. The integration of the telecommunications, broadcasting, and mobile Internet industries will provide consumers with a more convenient way to obtain music.

The era of wireless music (2009–present): The most noteworthy is that the popularity of mobile Internet has promoted the development of digital music. With the transformation of China's digital music industry from 2G, 3G to 4G, the value of music has been constantly refreshed. China's digital music market reached 1.79 billion yuan in 2009, with wireless music accounting for 92.1%. Wireless music has become an important revenue component of music content providers. The only problem is that the prevalence of piracy is not effectively controlled.

The “Notice on Ordering Online Music Service Providers to Stop Unauthorized Dissemination of Music Works” issued by the National Copyright Administration on July 2015 is considered as a key step in the standardization of the Chinese music copyright market. As of 2018, the size of China's digital music market was 7.63 billion yuan, with a growth rate of 113.2%. Digital music users reached about 700 million people, and the number of paying users was nearly 40 million. China's digital music market has formed a “one super, many strong” pattern. The platforms have begun to achieve a monopoly situation from the past to the start of mutual authorization. Strengthening the production of upstream content and expanding diversified profit channels have become development priorities and business models are gradually mature.

2 Optimization of China Digital Music Industry Structure Under 5G Network

On November 14, 2017, the Ministry of Industry and Information Technology of China issued a 5G system frequency usage plan, which identified 3300–3400 MHz, 3400–3600 MHz, and 4800–5000 MHz frequency bands as China's 5G working frequency bands [1]. Compared with 4G, 5G forms a network system that integrates multiple technologies and multiple services through the characteristics of millisecond-level delay, high reliability, high speed, and large bandwidth, which can provide more extensive user groups with high-quality services, and will to a large extent promote the optimization and upgrading of the digital music industry structure and the integration and development of related industries [2] (see Table 1 for details).

Table 1. 2G-5G network performance indicators

Name	Peak rate	User experience rate	Connection density	System	Adapter
2G	Peak data rate ≥ 30 kbps	User experience rate ≥ 15 Kbps	Connection density 100 users/km ²	GSM, CDMA	Voice interaction, text messaging
3G	Peak data rate ≥ 2 Mbps	User experience rate ≥ 300 kbps	Connection density 1000 users/km ²	TD-SCDMA, WCDMA, CDMA2000	Meeting general quality audio and video transmission
4G	Peak data rate ≥ 100 Mbps	User experience rate ≥ 20 Mbps	Connection density 2000 users/km ²	TD-LTE, FDD-LTE	Meeting high-quality audio and video transmission
5G	Peak data rate ≥ 1 Gbps	User experience rate ≥ 100 Mbps	Connection density 1million users/km ²	NR/TD-LTE, GSM, LTEFDD, WCDMA	Internet of Everything: AI VR, AR etc.

In the 5G era, communication systems will gain the abilities to interact with the environment, and the targets are expanded to joint optimizations of ever-increasing numbers of key performance indicators (KPIs), including latency, reliability, connection density, and user experience [3]. It means that under the background of 5G network technology, users can realize Internet of everything in every scene of life with certainty. For the market, this is also a brand-new business model and a brand-new challenge.

2.1 5G-Based Artificial Intelligence Composition Technology

As the 5G technology and big data and cloud computing are continuously applied, artificial intelligence composition technology has made great progress in the fields of music education, music creation, music performance and entertainment services. Artificial Intelligence Composition, originating from “algorithmic composition”, is an attempt to use a formal process to enable people (or composers) to use computers to achieve different degrees of automation in music composition [4, 5]. At present, computer-aided algorithm composition systems developed abroad such as Super Collider, C Sound, MAX/MSP, Kyma, Nyquist, AC Toolbox, etc.

Domestic algorithms are mainly artificial neural network systems (an algorithmic mathematical model that imitates the behavioral characteristics of biological neural networks and performs distributed parallel information processing), genetic algorithm (a global optimization algorithm that uses adaptive functions to evolve samples) and hybrid algorithm (combination of genetic algorithm and artificial neural network systems) [6]. For the compositions formed based on the large-scale large data material library and with the combination of these algorithmic composition technologies with artificial intelligence composition technology and the artificial intelligence composition system it is almost impossible to distinguish between works created by artificial or intelligent composition systems.

The most noteworthy thing is that the artificial intelligence composition system can create user-owned songs based on pictures, lyrics or a melody provided by the user, which is actually a process of user generated content (UGC). Users can upload information through the 5G network to the cloud, and then use the artificial intelligence composition system to convert the music works that users need (see Fig. 1).

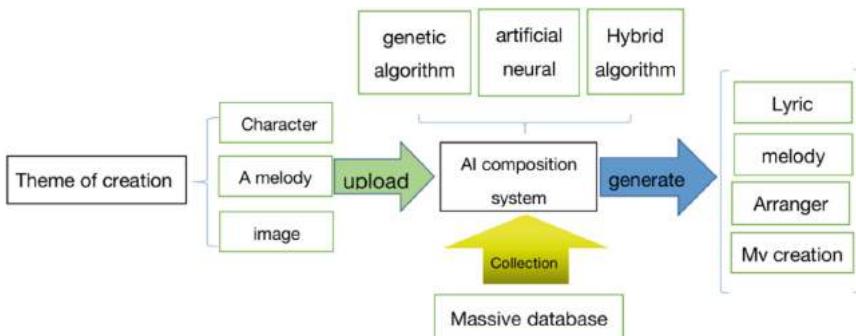


Fig. 1. The process of creating songs by artificial intelligence composition system

With the advent of the music composition system with artificial intelligence, audience without the theoretical knowledge of music composition can readily create their representative works. Users only need to prepare images, melodies or other relevant materials and upload to the AI music composition system; upon receiving instructions, the system will fulfill the creation of lyrics, music composition, arrangement and even the creation of the music video through composition methods such as genetic algorithm,

artificial neural network and so on with the incorporation of big data. These processes are completed in an instant, which are beyond imagination in the past.

Furthermore, as of December 2020, the number terminals with established 5G connection has exceeded 200 million, and 5G mobile phone shipments have exceeded 144 million units. In 2020, 580, 000 units of 5G base stations have been increased, and all cities have basically realized 5G network coverage [7]. 5G networks will tailor the provisioning mechanisms for more applications and services, which makes it more challenging in terms of complicated configuration issues and evolving service requirements [8]. Specifically, the changes in the music market will reflect the following characteristics:

- **The Feasibility of Grassroots Music Creation**

In the past, music creation was generally fulfilled by professional composers, but nowadays it is difficult for people to distinguish between works created by artificial intelligence music composition systems and works created by musicians, indicating that the technology has gradually matured. In addition, although consumer applications of 5G are still in the initial stage of introduction, with the gradual advancement of the 5G market in the future, such an enormous user group, it will inevitably give birth to a series of innovative applications, which would actively promote 5G powered video entertainment applications, and it is necessary to develop the layout of consumer applications through cooperation with Internet companies, including AR/VR live broadcasting, etc. In the future, with the popularization of 5G networks and artificial intelligence chips, developers will vigorously promote the marketization of artificial intelligence composition systems and the research and development of related mobile applications, making it possible for people to realize the previously unattainable music composition work an easy entertainment.

- **Everyone can become a Copyright Receptor of Music Works**

As blockchain technology matures, the integration of 5G network technology and blockchain technology will make the music market more transparent and diversified for earning. With the assistance of the secure and reliable database of blockchain and the highly decentralized and liberalized management mode, copyright maintenance operation of music works can be greatly reduced and the transaction of music products can be facilitated. On the 2019 Smart China Expo, Blockchain and Digital Economy Joint Laboratory of China Telecom released “The White Paper on Blockchain Smart Phones in the 5G Era”, in which it is described a blockchain application ecology established by China Telecom, which would lead the new change of digital economy with its distinctive decentralization Technology. It provides a safer hardware infrastructure for 5G mobile phones, makes five mobile phones become nodes of decentralized operation and data, so as to form a new value transmission network and establish a more secure ecosystem for mobile phone users [9]. In this way, the practical utility of artificial intelligence composition technology will be brought to the ultimate. For example, under the background of blockchain, ordinary users can use artificial intelligence music composition platforms to create their own works, and then released to the blockchain platform supported by 5G architecture, with its independent digital wallet and smart contracts (transparency for transactions between music performers, producers and fans) has a great market advantage, for example, a user publishes a new work to the platform, once a consumer has purchased the copyright of the work,

the funds will be automatically distributed to the accounts of creators, propagandists, performers and so on according to a preset proportion.

- **It will Replace Human Musical Activities to Some Extent**

In September 2018, Google held a large-scale AI interactive experience exhibition on Shanghai Long Museum, where the audience can interact with the AI robots through language, dancing or graffiti. What's amazing is that the artificial intelligent robots can extemporaneously play with the performers when the audience sings or plays at will. Due to the powerful composition technology of artificial intelligence, it will be favored by many performing arts groups and performing arts companies in the future, and in anticipation, it will probably replace part of music technicians who are active in the current market. Firstly, once the artificial intelligence composition technology matures and is launched to the market, it will have a great impact on the industries of music performance, music creation and music production. To cite an obvious example, during the days when the MIDI composition technology has not been invented, if a singer needs to use orchestral accompaniment while performing a song, generally, it would need a large instrumental ensemble composed of string instruments, wind instruments and percussion instruments, which would be at least 20 to 30 people. However, with the introducing of MIDI composition technology, with only a professional MIDI composer in front of the computer, recording of all the sound making and instrument play could be handled, the cost would be greatly reduced, and very little time and effort have to be spent. Similarly, nowadays, the emergence of artificial intelligent composition technology will also bear part of the work of composers; not only that, behind the stage of performance, some of the work will be streamlined due to this technology, and not so many messy synthesizers, effectors and other relevant electronic composition equipment would be needed, which would mark another technical change in the history of music technology.

2.2 Virtual Concert and AR Augmented Reality Experience

The 2019 World VR Industry Conference with the theme of “VR makes the world a better place—VR + 5G opens a new era of perception” was held in Nanchang, Jiangxi Province on October 19, 2019. Guo Ping, Huawei's rotating chairman, said in a speech: “VR/AR, as the first application in the 5G era, is a revolution in human-computer interaction and a revolutionary upgrade in computing power, connectivity and display.” AR (Augmented Reality) is a relatively new technology that promotes the integration between real-world information and virtual world information content. It is a virtual reality technology developed in the 20th century. The requirements for panoramic real-time and calculation are extremely strict. Not only does real-time recognition and automatic tracking of real scenes be required, but also it consumes ultra-high bandwidth. Under traditional 4G networks, VR and AR cannot fully reflect its value. However, as the 5G commercial process continues to advance, new opportunities will inevitably be ushered in [10].

Virtual concert: Faye Wong's “Magic Music” concert on December 30, 2016 was the first time that VR was used for live webcasts in China, but the coverage of 4G network of the time was unable to meet the massive data transmission requirements of VR and it was difficult to maintain its stability. In general, the live broadcast of music concerts is mainly guaranteed through wired networks. However, VR uses 5G to connect 1 million

users per square kilometer with massive connections, low latency and extremely fast features, so that fans can easily watch immersive live broadcasts at home [11].

Minecraft games also recently hosted the Fire Festival 2019. This is a live virtual music festival held entirely in the game. Over 50 artists performed at the event, and 6,500 people participated in the live interaction of the festival. The bars, pools, sea cliffs, private parties appearing in the game are virtual, but the audience has real experience in the virtual game world. In addition, 5G + VR opens up a lot of possibilities for musicians. It is not necessary to be in a physical space to co-create the same piece of music. The performer can accompany concert singers at different places, etc., which is enough to widen the horizon of the entire music industry.

AR mobile augmented reality experience: High-speed, stable, low-latency 5G mobile communication network, deep integration with Wi-Fi, allowing digital music to be online at any time, the fusion of reality, augmented reality, augmented virtual and virtual reality blurs the boundaries between the virtual world and the real world. Fans can experience the story behind the song and the music MV through VR. Mixing the real world allows fans to deeply experience the story and content of the song in the creation of songs across time and space, be immersed in the three-dimensional virtual world, and truly delineate a new field of virtual reality mapping for the viewer [12].

2.3 5G-Based Music Blockchain

Blockchain technology refers to the technical solution of collectively maintaining a reliable database through decentralization and trustfree. One of its biggest features is decentralization. The entire block is an open source system without unified database management organization, and everyone can edit the data. In addition, with the application of distributed ledger technology, everyone will back up the same information, which will inevitably form a huge database. Traditional 4G technology cannot undertake the massive data transmission tasks, while 5G has more extreme experience and larger capacity and uses the radio frequency spectrum to transmit massive amounts of data with higher speed and reliability than previous technologies. It can be therefore estimated that 5G will essentially help blockchain technology to create a new three-dimensional digital environment.

Blockchain technology and 5G technology, once applied in the music field, will break the traditional profit model and form a new music copyright trading platform and an efficient, free and open network environment. With the support of 5G technology, the blockchain will solve a series of problems such as the opacity of copyright holders' benefits, long benefit periods, and no rights for the copyright holders to speak. Without the "intermediary agency", the copyright owner does not need to deal with complicated intermediaries, and will get the maximum benefit. In addition, the virtual wallet (virtual digital cryptocurrency), smart contracts (transparency between musicians and fans), and distributed databases (storing all relevant copyright databases) of blockchain technology will form a new copyright transaction platform. Piracy and infringement will be significantly reduced.

3 In-Depth Development of Digital Music Industry Convergence Format under 5G Network

Under the traditional 4G network technology, blockchain, 3D, VR, and AR technologies have made certain progress. The arrival of 5G is bound to accelerate the intelligent layout of human beings in the field of life, and to make smart interconnection, smart home, and smart travel a new normal. Under 5G technology, digital music will work hand in hand with intelligent technologies such as the Internet of Things, artificial intelligence, and cloud computing to create a new way of life and work for humanity.

- **Smart Home**

5G network technology has become an era label. It will drive the rapid layout and development of the smart home industry, foster new economic drivers, trigger profound changes in production methods and lifestyles, accelerate the digitization, networking, and intelligence of production activities and promote the transformation and development of the real economy [13]. Digital music will appear in more scenes in the field of smart home. The mobile client is used to personalize and modulate scene modes in different time periods and the corresponding music style. For example, in the early morning, you can modulate a relaxed and cheerful music scene, and when you are holding a small party with friends, you can modulate a lively music scene and so on. All this will be completed independently by the artificial intelligence home system. The deep integration of digital music and smart home systems not only promotes the rapid and comprehensive development of the smart home industry, but also changes our lives in a completely new way.

- **Transportation**

Vehicle to Everything (V2X) is an information technology to connect vehicles with everything. V2X technologies attract industrial and academic efforts to provide wireless connectivity between all road entities and support several V2X services [14]. Under the 5G network technology, the use of on-board FM and on-board CD users will decline, and wireless digital music will be more and more popular, which will greatly promote the development of the digital music industry. With the development of the Internet, the year-on-year increase in car ownership, and the youthfulness of car buyers, consumers are no longer able to meet the basic security functions of the Internet of Vehicles, and music overhearing entertainment has become the first demand for "Internet of Vehicles". Therefore, the urgency of consumer and vehicle terminal manufacturers' demand for in-vehicle audio content has also increased rapidly.

- **Medical Field**

A large amount of previous clinical experience has shown that music therapy can treat human diseases from the physical and psychological aspects, and has significant effects on poor thinking activities, lack of will, lack of interest, and attention disorders. For example, to relieve the anxiety of cancer patients, especially the treatment of patients with chronic schizophrenia supplemented with music therapy on the basis of antipsychotic drugs will greatly improve the patient's condition. The intelligent system can monitor real-time according to people's sleep, exercise, heart rate, etc. and adjust through intelligent terminals. In July 2017, the State Council issued the "New Generation Artificial Intelligence Development Plan", which proposes that by

2030, China's AI theory, technology and applications will generally reach world-leading levels. The continuous integration of digital music, artificial intelligence and medical treatment is bound to promote major changes in the human medical field.

4 Summary

The fusion of 5G, Internet, Internet of Things, and blockchain technologies, especially with virtual reality, augmented reality, mixed reality, robotics and other technologies, will greatly change the way humans watch the performances. In addition, the continuous maturation of artificial intelligence composition technology will eventually realize the possibility of creating music at the grassroots level, that is, everyone can complete a more personalized musical composition through the artificial intelligence composition system. The deep integration of digital music with smart home, medical, and connected cars will let us see that under the blessing of 5G, human production methods and lifestyles are moving towards digitalization, networking, and intelligence.

Of course, the realization of technology is just a prerequisite. To truly realize human digitization, networking, and intelligence, what we still need is the technology based on the development of social productivity and the improvement of creativity. In the 5G era, first of all, we should attach great importance to the construction of a talent team for technology production and R & D, which is an important guarantee for China to fully enter the 5G era. Second, productivity and the market are needed as the guarantee. The fundamental way to enable the digital market to transform and develop in the 5G era ultimately depends on the market. Finally, the government and enterprises need to play a leading role and construct a good system and institutional environment to make necessary guarantees.

References

1. Zhao, J., Sun, Y., Liu, S.F.: Research on 5G network deployment scheme. *China New Telecommun.* **9**(10), 2–7 (2018)
2. Liu, C.F.: How 5G will rewrite the media industry. *Media* **3**(06), 6–7 (2019)
3. Yang, W.J., Wang, M., Zhang, J.J., et al.: Narrowband wireless access for low-power massive internet of things: a bandwidth perspective. *IEEE Wirel. Commun.* **24**, 138–145 (2017)
4. Gareth, L., Curtis, A.: Programming languages for computer music synthesis, performance, and composition. *ACM Comput. Surv.* **17**(2), 235–265 (1985)
5. Baratè, A., et al.: 5G technology for music education: a feasibility study. *IADIS Int. J. Comput. Sci. Inf. Syst.* **14**(1), 31–52 (2019). ISSN 1646-3692
6. Zhou, L., Deng, Y.: Research on the status quo and trend of the development of artificial intelligence composition. *Arts Explor.* **32**(05), 107–111 (2018)
7. NetEase News. <https://3g.163.com/news/article/FV5P1HJU075497TJ.html>. Accessed 31 Dec 2020
8. SOHU. https://www.sohu.com/a/298792526_114949. Accessed 10 Sept 2019
9. You, X.H., Zhang, C., Tan, X., Jin, S., Wu, H.: AI for 5G: research directions and paradigms. *Sci. China Inf. Sci.* **62**(2), 1–13 (2018). <https://doi.org/10.1007/s11432-018-9596-5>
10. Ge, X., Pan, L., Li, Q., Mao, G., Song, T.: Multipath cooperative communications networks for augmented and virtual reality transmission. *IEEE Trans. Multimed.* **19**(10), 2345–2358 (2017). <https://doi.org/10.1109/TMM.2017.2733461>

11. Baratè, A., et al.: 5G technology for augmented and virtual reality in education. In: Conference: International Conference on Education and New Developments (2019)
12. Qiao, X., Ren, P., Dustdar, S., et al.: Web AR: a promising future for mobile augmented reality—state of the art, challenges, and insights. In: Proceedings of the IEEE, vol. 107, no. 4, pp. 651–666, April 2019. <https://doi.org/10.1109/JPROC.2019.2895105>
13. Chen, R.D.: 2019 digital market in 5G era. Spec. Plan. **11**(04), 18–19, 1004–3381 (2019)
14. Abdel, S.A., Hakeem, A.A., Hady, H.K.: Current and future developments to improve 5G-NewRadio performance in vehicle-to-everything communications. Telecommun. Syst. **75**(3), 331–353 (2020). <https://doi.org/10.1007/s11235-020-00704-7>



Optimal Solution of Transportation Problem with Effective Approach Mount Order Method: An Operational Research Tool

Mohammad Rashid Hussain^(✉), Ayman Qahmash, Salem Alelyani,
and Mohammed Saleh Alsaqer

Center for Artificial Intelligence (CAI), College of Computer Science, King Khalid University,
P.O. Box 960, Abha 61421, Kingdom of Saudi Arabia
humhammad@kku.edu.sa

Abstract. This paper introduces a new approach to optimize the cost per unit of product for the Transportation Problem to achieve better outcomes. We present Basic Feasible Solution (BFS) approach compromised of five main steps: (1) Create a Matrix A = mod |Supply(s_i)-Demand(d_j)| (2) Add the cost of each cell of cost matrix C with corresponding elements of Matrix A and Create Matrix B. (3) Mark number in Ascending order of each elements of Matrix B from 1 to mxn (4) If $s_i \neq d_j$, then $\zeta = \text{lsmall}(s_i, d_j)$, else $\zeta = s_i$ or d_j . Assign ζ in Matrix B to smallest number from 1 to mxn, and cut the rest of elements of row ζ or column ζ and subtract ζ from other than selected; and (5) Repeat step 4, until all the supply and demand become zero. This solution approach finds the basic feasible solution of TP with the same complexity for solving Vogel's approximation method (VAM).

Keywords: Mount Order Method (MOM) · Transportation Problem (TP) · Basic Feasible Solution (BFS) · Optimal Solution (OS) · Stepping stone method

1 Introduction

A set of non negative individual allocation which satisfies all the given constraints is termed as feasible solution. Basically in linear programming more importance is about basic feasible solution rather than basic solution (Solution of a problem which satisfies all the condition). Basic feasible solution (BFS) - For $m \times n$ matrix transportation problem, if numbers of allocations are $m + n - 1$ then it is called as basic feasible solution. To compute the Basic Feasible Solution (BFS) of Transportation Problem (TP), the efficient method which has been observed is Vogel's approximation method (VAM) than other existing method, which results nearest to optimal solution. In comparative study, the performance of proposed Mount Order Method (MOM) has been compared with VAM to prove the effectiveness of proposed one. Operational research (OR) is a mathematical method, and the scientific study of OR make a better decision. It offers powerful tools to analyze and understand certain classes of problems to achieve better outcomes. Optimization problem (OP) identifies the values of the decision variables that

lead to the best outcomes for the decision maker within the assumption of the model. Optimization and a feasible solution is an optimal solution (OS) which maximize or minimize the measure of goodness. The solution to a transportation problem (TP) has two stages; one is called the identifying a basic feasible solution (BFS) and the second are to get the OS. A set of options that result in the conditions are satisfied is called BFS; this paper presents an approach to find a BFS of TP, with a better result towards the existing approach. We first define the existing approach and suggested one intelligent approach that it can give better result compare to the existing method. A brief review of the existing method of TP and subsequent approach follows. We then offer a selective and better approach which TP is used to identifying a BFS and provide some evidence by comparing with the existing approach. The proposed approach suggests, how operations researchers approach new problems, provide a brief survey of different techniques which have been developed by the researchers of OR. Operation researchers are doing research in their area of interest to achieve better outcomes. The OS identifies the values of the decision variables that lead to the best outcomes for the decision maker within the assumption of the model. The solution to a TP takes about transporting a single item from a given set of supply point to a given set of the destination point. In Table 1 and Table 2, the supply in supply point i is s_i and the requirement in-demand point or destination point j is d_j . The problem is finding the least cost transportation from the supply points to demand points where, unit cost c_{ij} of transportation. The problem is formulated as x_{ij} which is the product transported from supply point i to demand point j .

Table 1. Transportation tableau: quantity transported from supply point i to demand point j .

	D_1	D_2	...	D_j	
S_1	c_{11} x_{11}	c_{12} x_{12}	...	c_{1j} x_{1j}	a_1
S_2	c_{21} x_{21}	c_{22} x_{22}	...	c_{2j} x_{2j}	a_2
...
S_i	c_{i1} x_{i1}	c_{i2} x_{i2}	...	c_{ij} x_{ij}	a_i
	b_1	b_2		b_j	

The given set of supply from supply point i is a_i and the requirement at demand point j is b_j and to find the least cost solution of the problem, where c_{ij} is the per unit cost of product.

The quantity of supply product given from point i to required point j is the problem finalized as x_{ij} . To minimize the total cost of a product $c_{ij}x_{ij}$ is the objective function of subject to sets of constraints.

$$\sum a_i = \text{Total availability, and } \sum b_j = \text{Total requirements.}$$

If $\sum a_i \geq \sum b_j$, and $c_{ij} \geq 0$: Total availability > the requirements, i.e. the possible to fulfill all requirements of the different set of product).

If $\sum a_i < \sum b_j$, and $c_{ij} \geq 0$: Total availability < the requirements (then clearly all the requirements cannot be met).

To minimize the cost of the transportation $c_{ij}x_{ij}$ is the objective function of subject to supply constraints. For every supply point, the total quantity that leads the point should \leq what is available. s_i is the quantity which is available in the supply point i and as far as every demand point or destination point j is d_j concerned.

x_{ij} : quantity transported from i to j.

$$\text{Minimize } \sum \sum c_{ij}x_{ij}$$

$$\sum x_{ij} \leq s_i$$

$$\sum x_{ij} \leq d_j$$

$$x_{ij} \geq 0, c_{ij} \geq 0, \forall i, j$$

TP is a slightly different version of the simplex algorithm.

$$\sum s_i = \text{Total availability, and } \sum d_j = \text{Total requirements.}$$

If $\sum s_i \geq \sum d_j$, Which means total availability is more than what is required (possible to transport the entire requirement such that the demand of every destination point is met). So it is possible to minimize the transportation cost. Transportation cost cannot be negative $c_{ij} \geq 0, \forall i, j$. We will only end up sending exactly the amount that is required and none of these destination points will receive even one unit more than it requires because the extra unit has to be transported from one of these supply points and that can only increase the cost of transportation. Cost optimization in the age of digital business means that organizations use a mix of IT and business cost optimization for increased business performance through wise technology investments”, says John Roberts, research vice president, and distinguished analyst with Gartner’s CIO and Executive Leadership team [6]. “The key to effective enterprise cost optimization is to have proactive processes in place as part of business and technology strategy development to continually explore new opportunities (John Roberts). In recent year OR is playing a major role in the different aspect of healthcare. It is optimizing radiation treatment for cancer, allocating the resources for HIV prevention programs and so on. The concept of getting the feasible solution and its variants has applications in many areas such as development, optimization, planning, and design making application of Operation Research models for public health in healthcare are discussed in [1] and the clinical

problems are discussed in [3], to optimize a system, and software facilitates are the problem design methods. The Queuing theory of operation research concept has been applied to reduce the patients waiting time through Monte Carlo modeling method [2, 12] used Total Opportunity Cost Method (TOCM) and find out the distribution indicator for each cell of the TOCM through the respective element, subtracting corresponding row and column highest element of each cell and the cell having smallest distribution indicator (SDI) are allocated [4]. Based on the TOCM Approach, some of the methods have been developed like TOCM-MMM, TOCM-VAM, TOCM-EDM, TOCM-HCDM, and TOCM-SUM, etc. [7–9] in 2012. Finding BFS is a prior requirement to obtain an optimal solution [5], some standard method to find BFS are NWC, LCM and VAM and one another method also have been developed by Ahmed et al. [10] in 2016 is called Allocation Table Method (ATM). Some of the improvement is basically achieved by making changes in the available solution procedure of the classical LCM [11]. We observed that VAM is one of the best existing BFS methods that as a follow-up procedure to get an optimal solution, this method is based on cost cell approach, and NWC is based on the position that does not mean about per unit cost of the product. Our aim is to compare our approach with the best existing method to prove the best performance of approach to solving the transportation problem to find the basic feasible solution, which results near to an optimal solution.

2 Problem Description

Based on the results achieved by different methods are compared and discussed. The validity and utility of the LP model can be the judge to justify the performance of the proposed method. Validation by results consists of a comparison of different methods and their solution with corresponding outcomes. A comparative study between existing and proposed methods are carried out by solving some numbers of transportation problems that shows the proposed approach gives a better solution in comparison to the existing method.

2.1 Basic Feasible Solution (BFS) of TP

A BFS can be found by

- Northwest corner Rule
- Least Cost method
- Vogel's approximation method
- Proposed method: Mount order method

A BFS to a TP Satisfies the Following Conditions

- i. The row-column (supply-demand) constraints are satisfied.
- ii. The non-negativity constraints are satisfied.

- iii. The allocation is independent and does not form a loop.
- iv. There is exactly $m + n - 1$ allocation.

All the existing and proposed methods are satisfying the first two conditions.

- A loop can be broken in two ways, one of which will result in a solution with lesser (non-increasing) cost. It is advisable to break the loop and obtain a better solution with lesser cost.
- A feasible solution has more than $m + n - 1$ allocation, there is a loop and we break it to get a basic feasible solution without a loop.
- We observed, all the methods (existing and proposed) will not give more than $m + n - 1$ allocation.
- An exact $m + n - 1$ allocation also does not guarantee the loop (either exist or not), there might be the hidden loop.
- A degenerate basic feasible solution has fewer than $m + n - 1$ allocation. Introduce a ϵ such a way in which it does not form a loop.

2.2 Optimal Solution to a TP

There are two different methods that we use to find an optimal solution.

- Stepping Stone Method
- Modified distribution method (MODI) or U-V Method

These methods used to determine optimality of a basic feasible solution (i.e. North-west Corner Rule, Least Cost or Vogel's Approximation, and proposed "Mount order method").

Optimization and a feasible solution is an optimal solution (OS) which maximize or minimize the measure of goodness. The optimal solution sends exactly the quantity that is needed in each of these destination points.

1. if $\sum s_i = \sum d_j$, (Balanced transportation problem), it will meet the exact requirement of every destination point. It means that inequality becomes an equation $\sum x_{ij} = s_i, \forall i$, and $\sum x_{ij} = d_j, \forall j$.
2. If the condition is not balanced (unbalanced), then need to convert to make the condition balanced and solve it.
 - (i) if $\sum s_i \leq \sum d_j$, The entire requirement cannot be met.
 - (ii) if $\sum s_i \geq \sum d_j$, the entire demand cannot be met.

Optimization is a synonym of mathematical programming for which computer programming is used to implement. It is an economy of compactness and predictive power, correctness, and coherence, in which the problem has discussed to represent a real-world situation. Optimization and a feasible solution is an OS which maximizes or minimizes the measure of goodness. The linear programming (LP) may or may not have a feasible solution. If feasible region is empty then LP does not have a solution. If the solution exists,

means the feasible region is non-empty but the condition of bounded is not depended on the region. Optimization problem identifies the values of the decision variables that lead to the best outcomes for the decision maker within the assumption of the model. Transportation problem is a slightly different version of the simplex algorithm. The solution to a TP has two stages, one is called identifying a basic feasible solution and the second is to get the optimal solution.

Stepping Stone Method

This is one of the methods used to determine optimality of an initial basic feasible solution. Finding a BFS is the prior requirement to obtain an optimal solution for the transportation problem. There are three standard BFS methods have been discussed in all standard books (Hillier and etc.) and these are Northwest Corner Rule (NWC), Least Cost Method (LCM), Vogel's Approximation Method (VAM). In this paper, a new approach "Mount Order Method" (MOM) is proposed to obtain a BFS for the TP. This method tries to enter every one of the unallocated position under the assumption that present solution is not optimal then at least one of this unallocated position should have an allocation. So this method tries to it exhaustively look at a decrease (because of putting +1 in the unallocated position and completing the loop). It enters that and tries to move by reducing the objective function further. Now, at the optimum when we unable to find an entering unallocated position. We say that the optimum is reached.

3 Solution Approach

Representation of Transportation Tableau

Table 2. Unit cost transportation tableau

c_{ij} : Per unit cost of transportation, where $c_{ij} \geq 0$ for all i and j , $\forall i, j, i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$					
Demand →	$(D_J)_{J=1}, \forall i$	$(D_J)_{J=2}, \forall i$	$(D_J)_{J=3}, \forall i$...	$(D_J)_{J=n}, \forall i$
Supply↓	$(S_i)_{i=1}, \forall j$	$(c_{1j})_{j=1}$	$(c_{1j})_{j=2}$	$(c_{1j})_{j=3}$...
	$(S_i)_{i=2}, \forall j$	$(c_{2j})_{j=1}$	$(c_{2j})_{j=2}$	$(c_{2j})_{j=3}$...
	$(S_i)_{i=3}, \forall j$	$(c_{3j})_{j=1}$	$(c_{3j})_{j=2}$	$(c_{3j})_{j=3}$...
...
$(S_i)_{i=m}, \forall j$	$(c_{mj})_{j=1}$	$(c_{mj})_{j=2}$	$(c_{mj})_{j=3}$...	$(c_{mj})_{j=n}$

$$(S_i, D_j : c_{ij}) \rightarrow \text{Set of supply and demand with per unit cost}$$

$x_{ij} \rightarrow$ Quantity transported from supply point i to demand point j.

$$S_i \rightarrow (a_i)_{i \rightarrow 1 \text{ to } m}$$

$$S_i : (a_i)_{i \rightarrow 1 \text{ to } m} \rightarrow (a_1, a_2, a_3, \dots, a_m) \quad (1)$$

$$\left. \begin{array}{l} \text{Per unit cost of Supplied product } a_i \text{ Supply point } j \\ (S_i)_{i=1} \rightarrow (c_{1j})_{\forall j} \\ (S_i)_{i=2} \rightarrow (c_{2j})_{\forall j} \\ \cdot \\ \cdot \\ (S_i)_{i=m} \rightarrow (c_{mj})_{\forall j} \end{array} \right\} j = 1 \text{ to } n$$

$$\left. \begin{array}{l} (S_i)_{i=1, \forall j} \rightarrow (c_{1j})_{j=1}, (c_{1j})_{j=2}, (c_{1j})_{j=3}, \dots, (c_{1j})_{j=n} \\ (S_i)_{i=2, \forall j} \rightarrow (c_{2j})_{j=1}, (c_{2j})_{j=2}, (c_{2j})_{j=3}, \dots, (c_{2j})_{j=n} \\ (S_i)_{i=3, \forall j} \rightarrow (c_{3j})_{j=1}, (c_{3j})_{j=2}, (c_{3j})_{j=3}, \dots, (c_{3j})_{j=n} \\ \cdot \\ (S_i)_{i=m, \forall j} \rightarrow (c_{mj})_{j=1}, (c_{mj})_{j=2}, (c_{mj})_{j=3}, \dots, (c_{mj})_{j=n} \end{array} \right\} \quad (2)$$

$$D_J \rightarrow (b_j)_{j \rightarrow 1 \text{ to } n}$$

$$D_J : (b_j)_{j \rightarrow 1 \text{ to } n} \rightarrow (b_1, b_2, b_3, \dots, b_n) \quad (3)$$

$$\left. \begin{array}{l} \text{Per unit cost of Demanded product } a_i \text{ demand point } j \\ (D_J)_{J=1} \rightarrow (c_{i1})_{\forall i} \\ (D_J)_{j=2} \rightarrow (c_{i2})_{\forall i} \\ \cdot \\ \cdot \\ (D_J)_{j=n} \rightarrow (c_{in})_{\forall i} \end{array} \right\} i = 1 \text{ to } m$$

$$(D_J)_{J=1 \text{ to } n} \rightarrow \left[\sum c_j, \sum x_j \rightarrow b_j \right] \rightarrow (c_{ij})_{\forall i, j=1 \text{ to } m} \quad (4)$$

$$\left. \begin{array}{l} \text{Minimize } \sum_{j=1}^n c_j x_j \\ \text{Subject to } \sum_{j=1}^n a_{ij} x_j \geq b_j, \text{ for all } j \\ x_j \geq 0, \text{ for all } j \end{array} \right\} \quad (5)$$

Algorithm: Mount Order Method (MOM)

Step 1. Create a Matrix A = mod | $s_i - d_j|$.

```
for (i = 1; i < m; ++ i); // m = No. of Rows
```

```
for (j = 1; j < n; ++ j); // n = No. of Columns
```

```
{ A[i][j] = [|si - dj|];
```

```
} printf ("Result Matrix A") // subtract supply from demand or demand from supply, the  
difference must be positive.
```

Step 2. Add the cost of each cell of cost Matrix C with corresponding elements of Matrix A and create a Matrix B.

Step 3. Mark number in ascending order of each elements of Matrix B from 1 to mxn,

if some elements of Matrix B is repeated then mark all repeated elements with same numbers as repeated from 1 to mxn.

Step 4. If $s_i \neq d_j$,

$$\zeta = |\text{small}(s_i, d_j)|$$

else

$$\zeta = s_i \text{ or } d_j$$

Assign ζ in Matrix B to smallest number from 1 to mxn, and cut the rest of elements of row ζ or column ζ and subtract ζ from other than selected ζ .

Step 5. Repeat step 4, until all the supply and demand become zero.//

Repeat step 4 till all the supply and demand assigned to cell.

4 Numerical Example

Example-1 of Transportation Problem to Find basic Feasible and Optimal Solution

$S_i : (a_i)_{i \rightarrow 1 \text{ to } 3} \rightarrow (10, 15, 40)$ Using Eq. (1).

$D_j : (b_j)_{j \rightarrow 1 \text{ to } 3} \rightarrow (20, 15, 30)$ Using Eq. (3).

$$\left. \begin{array}{l} (S_i)_{i=1}, \forall j \rightarrow (c_{1j})_{j=1}, (c_{1j})_{j=2}, (c_{1j})_{j=3} : (\$2, \$2, \$3) \\ (S_i)_{i=2}, \forall j \rightarrow (c_{2j})_{j=1}, (c_{2j})_{j=2}, (c_{2j})_{j=3} : (\$4, \$1, \$2) \\ (S_i)_{i=3}, \forall j \rightarrow (c_{3j})_{j=1}, (c_{3j})_{j=2}, (c_{3j})_{j=3} : (\$1, \$2, \$1) \end{array} \right\} \text{Using Eq. (2)}$$

Step 1:

```
Create a Matrix A = mod |si - dj|
for (i = 1; i < 3; ++ i);
for (j = 1; j < 3; ++ j);
A[i][j] = [|si - dj|];
} printf ("Result Matrix A")
```

$$\text{Matrix } A = [A] = \begin{bmatrix} 10 & 5 & 20 \\ 5 & 0 & 15 \\ 20 & 25 & 10 \end{bmatrix}$$

$$\text{Matrix } C = [C] = \begin{bmatrix} 2 & 2 & 3 \\ 4 & 1 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Step 2:

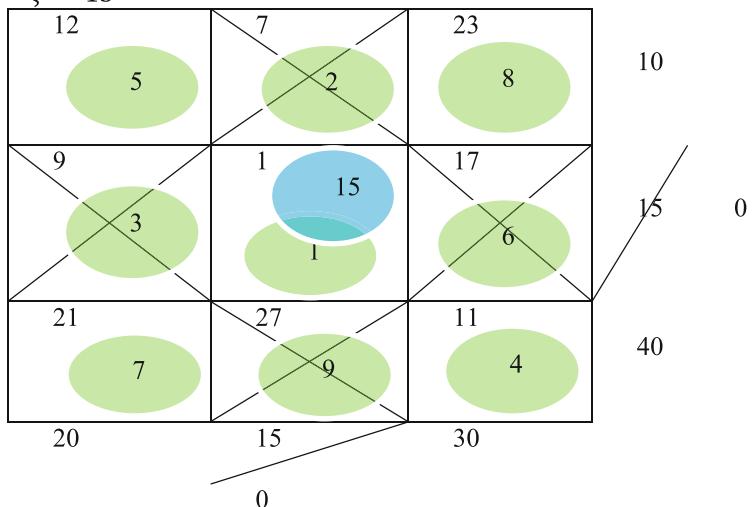
$$\text{Matrix } B = [B] = \begin{bmatrix} 12 & 7 & 23 \\ 9 & 1 & 17 \\ 21 & 27 & 11 \end{bmatrix}$$

Step 3:

12 5	7 2	23 8	10
9 3	1 1	17 6	15
21 7	27 9	11 4	40
20	15	30	

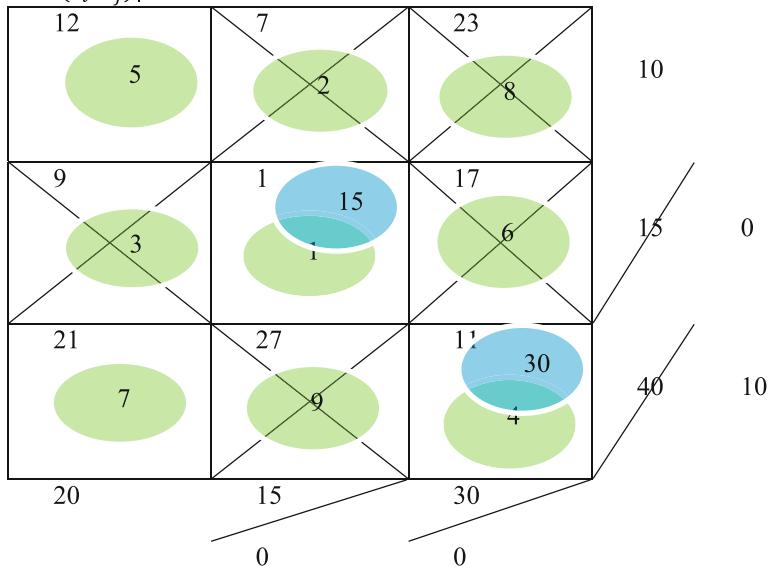
Step 4; $s_i = d_j = 15$

$$\zeta = 15$$



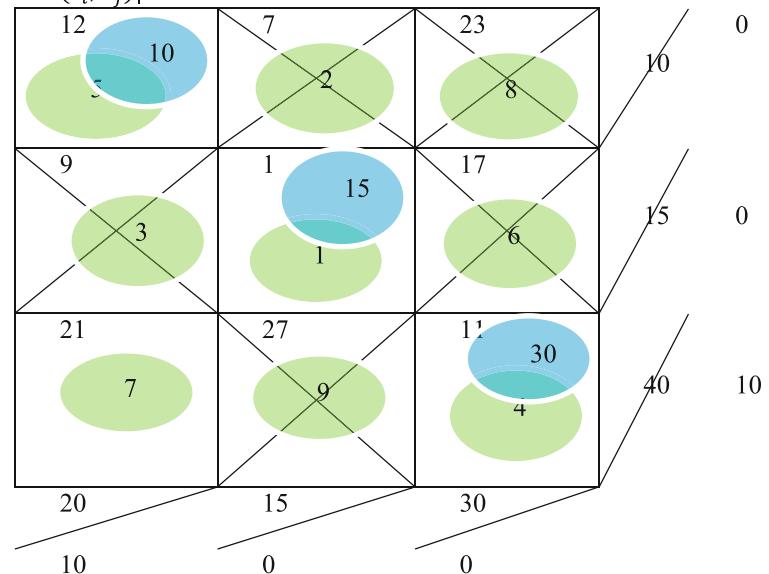
$$s_i \neq d_j, (40 \neq 30)$$

$$\zeta = |\text{small } (s_i, d_j)| = 30$$

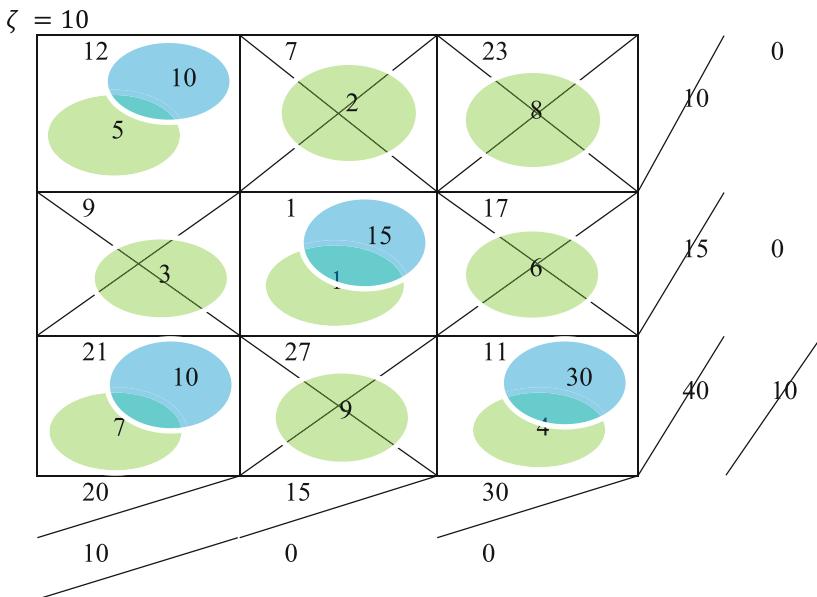


$$s_i \neq d_j, (10 \neq 20)$$

$$\zeta = |\text{small } (s_i, d_j)| = 10$$



$$s_i = d_j = 10$$



MOM Result

$$\left. \begin{array}{l} (D_J)_{J=1} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i1})_{i=1}, 10\} + \{(c_{i1})_{i=3}, 10\}] \\ (D_J)_{J=2} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i2})_{i=2}, 15\}] \\ (D_J)_{J=3} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i3})_{i=3}, 30\}] \end{array} \right\} \sum (D_J)_{J=1to3} = \$75$$

Existing methods (NWC, LCM, VAM)

Algorithm: North-West Corner Rule (NWC)

Step 1: Start in the upper left corner of the transportation table.

Step 2: Put x_{11} , $x_{11} = \min\{a_1, b_1\}$.

Step 3: if $x_{11} = a_1$, mark cross in row 1, no such more basic variables come from row 1 and put $b_1 = b_1 - a_1$.

Step 4: if $x_{11} = b_1$, mark cross in column 1, no more basic variables come from column 1 and put $a_1 = a_1 - b_1$.

Step 5: if $x_{11} = a_1 - b_1$, mark cross in either row 1 or column 1, but not both. put $b_1 = 0$, mark cross in row 1. Otherwise put $a_1 = 0$, mark cross in column 1.

Step 6: Carry on to apply this approach to the most north-west corner cell in the table which does not lie in a crossed-out row or column. Ultimately, there will be only one cell that may be assigned a value (Assign this cell value equal to its row or column demand, and mark cross in both the cells row and column).

Step 7: Now, BFS has been obtained

$$\left. \begin{array}{l} (D_J)_{J=1} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i1})_{i=1}, 10\} + \{(c_{i1})_{i=2}, 10\}] \\ (D_J)_{J=2} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i2})_{i=2}, 5\} + \{(c_{i2})_{i=3}, 10\}] \\ (D_J)_{J=3} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i3})_{i=3}, 30\}] \end{array} \right\} \sum (D_J)_{J=1to3} = \$115$$

Algorithm: Least Cost Method (LCM)

Step 1: Obtain cell with smallest $\{c_{ij}\}$.

Step 2: Put $x_{ij}, x_{ij} = \min \{a_i, b_j\}$ to cell of step 1.

If $x_{ij} = a_i$, mark cross row i, no more basic variables will come from row i and put b_j

$$= b_j - a_i.$$

If x_{ij}

$= b_j$, mark cross to the column j, no more basic variables will come from column j and put a_i
 $= a_i - b_j$.

If $x_{ij} = a_i - b_j = 0$, mark cross to either row i or column j, but not both. put $b_j = 0$, when mark cross to row i. Otherwise put $a_i = 0$, when mark cross to column j.

Step 3: Carry on to apply step 1 and step 2 with rest of the per unit cost element of transportation cost matrix.

Ultimately, only one cell that can be assigned a value (Assign this cell value = its row or column demand, and cut out both the cells row and column).

Step 4: Now, BFS has been obtained

Least Cost Method (LCM)

$$\left. \begin{array}{l} (D_j)_{j=1} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i1})_{i=3}, 20\}] \\ (D_j)_{j=2} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i2})_{i=2}, 15\}] \\ (D_j)_{j=3} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i3})_{i=1}, 10\} + \{(c_{i3})_{i=3}, 20\}] \end{array} \right\} \sum (D_j)_{j=1 \text{ to } 3} = \$85$$

Algorithm: Vogel's Approximation Method (VAM)

Step 1: Find penalty (subtract smallest per unit cost from next to smallest per unit cost in same row or column)

Step 2: Assign the variable to least possible per unit cost of biggest penalty row or column, if penalty are ties then select randomly. Rearrange supply or demand as following:

put $x_{ij} = \min \{a_i, b_j\}$.

If x_{ij}

$= a_i$, mark cross to row i, no more basic variables will come from row i and put b_j

$$= b_j - a_i.$$

If x_{ij}

$= b_j$, mark cross to the column j, no more basic variables will come from column j and put a_i
 $= a_i - b_j$.

If $x_{ij} = a_i - b_j = 0$, mark cross to either row i or column j, but not both. put b_j

$$= 0, \text{ when cross out row i. Otherwise put } a_i$$

$$= 0, \text{ when mark cross to column j.}$$

Step 3: Carry on applying this procedure. Ultimately there will be only one cell that can be marked a value (Mark this cell value equal to its row or column demand, and cut out both the cells row and column).

Step 4: Now, IBFS has been obtained

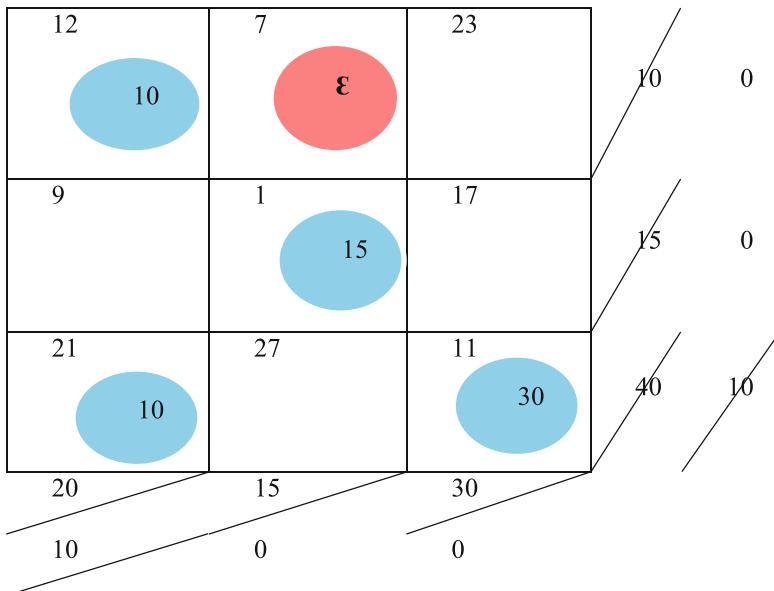
Vogel's Approximation Method (VAM)

$$\left. \begin{array}{l} (D_j)_{j=1} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i1})_{i=3}, 20\}] \\ (D_j)_{j=2} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i2})_{i=2}, 15\}] \\ (D_j)_{j=3} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [\{(c_{i3})_{i=1}, 10\} + \{(c_{i3})_{i=3}, 20\}] \end{array} \right\} \sum (D_j)_{j=1 \text{ to } 3} = \$85$$

Optimal Solution: Stepping Stone Method over MOM

- $x_{12}, x_{13}, x_{21}, x_{23}$, and x_{32} are unallocated cell or non-basic variables.

- Number of allocated cells are 4, which is less than $m + n - 1$ (degenerate feasible solution exist)
- Consider ε , to make the condition Basic Feasible Solution, select the position in which it must retain the true that these $m + n - 1$ position are independent and chose $\varepsilon = 0$ as an allocation
- Consider the position x_{12} for ε , which will not form a loop with allocated cells.
- Now, there are five non-basic position, put +1, one by one in each of the position, which makes a loop and evaluate the effect.
- Add +1 in any of unallocated cell, which makes a loop because $m + n - 1$ condition exceed.



Net cost (increased/decreased).

To look over the net cost by putting +1. Start from this unallocated cell with +1 and move to other allocated cell with an alternate sign to form a loop. Carry on this procedure one by one with all unallocated cell. The sign of the net cost will show that either net cost is increasing or decreasing is shown in Table 3.

There are four unallocated cells, putting +1 in each would give us an increased. So, we realized that none of them actually are capable of decreasing the objective function further from 75 to something lower than that, so, optimum is reached.

$$\left. \begin{array}{l} (D_J)_{J=1} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [(\{c_{i1}\}_{i=1}, 10) + (\{c_{i1}\}_{i=3}, 10)] \\ (D_J)_{J=2} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [(\{c_{i2}\}_{i=2}, 15)] \\ (D_J)_{J=3} \rightarrow [\sum c_j, \sum x_j \rightarrow b_j] \rightarrow [(\{c_{i3}\}_{i=3}, 30)] \end{array} \right\} \sum (D_J)_{J=1to3} = \$75$$

Hence, it proves the developed method is giving better result in comparison to the existing method.

Table 3. Net cost (increased/decreased): the sign of the net cost show that either net cost is increasing or decreasing.

Unallocated-cell	Loop of per unit cost cell	Net cost increased	Net cost decreased
x_{13}	$c_{13} - c_{33} + c_{31} - c_{11}$	1	-
x_{21}	$c_{21} - c_{22} + c_{12} - c_{11}$	3	-
x_{23}	$c_{23} - c_{33} + c_{31} - c_{11} + c_{12} - c_{22}$	1	-
x_{32}	$c_{32} - c_{31} + c_{11} - c_{12}$	1	-

5 Related Problem

Problem 1 is similar to all other rest of problems and can be solved by a similar approach.

Table 4. Supply-demand table of problem 1 to 6

Problem →	1	2	3	4	5	6
$S_i : (a_i)_{\forall i} \rightarrow$	10, 15, 40	40, 60, 50	40, 30, 30	22, 15, 8	15, 25, 10	200, 300, 500
$D_J : (b_j)_{\forall j} \rightarrow$	20, 15, 30	20, 30, 50, 50	20, 30, 50	7, 12, 17, 19	5, 15, 15, 15	200, 400, 400

Table 5. Per unit cost (in \$) of supply product table of problem 1 to 6

Problem →	1	2	3	4	5	6
$S_i : (c_{ij})_{i=1, \forall j} \rightarrow$	2, 2, 3	4, 6, 8, 8	2, 5, 2	5, 2, 4, 3	10, 2, 20, 11	2, 7, 5
$S_i : (c_{ij})_{i=2, \forall j} \rightarrow$	4, 1, 2	6, 8, 6, 7	1, 4, 2	4, 8, 1, 6	12, 7, 9, 20	3, 4, 2
$S_i : (c_{ij})_{i=3, \forall j} \rightarrow$	1, 2, 1	5, 7, 6, 8	4, 3, 2	4, 6, 7, 5	4, 14, 16, 18	5, 4, 7

In more of the cases, our proposed method is performing better than other methods and achieved result near to the optimal solution, which have been shown in Table 4, Table 5 and Table 6. Problems 1, 3, 4, and 6, BFS is exactly similar to the optimal solution, which is 75, 210, 104, and 3300. Table 7 shows that the proposed method performs better than another method (in %).

Table 6. Results of the proposed method are compared with existing methods of getting basic feasible solution and also compared with the optimal solution (in \$)

Problems	Basic feasible solution				Optimal solution Stepping stone method	
	Proposed method		Existing method			
	MOM	NWC	LCM	VAM		
1	75	115	85	85	75	
2	930	980	930	960	920	
3	210	280	270	210	210	
4	104	125	105	104	104	
5	475	520	475	475	435	
6	3300	4800	3300	3300	3300	

Table 7. Results of the proposed method are compared with existing methods of getting basic feasible solution and also compared with the optimal solution (in %)

Near about (in %) optimal solution			Problems					
			1	2	3	4	5	6
BFS	Proposed method	MOM	100%	98.92%	100%	100%	91.58%	100%
	Existing method	NWC	65.22%	93.88%	75%	83.2%	83.65%	68.75%
		LCM	88.24%	98.92%	77.78%	99.05%	91.58%	100%
		VAM	88.24%	95.83%	100%	100%	91.58%	100%

Table 8. Results of the proposed method are compared with existing methods of getting matched with optimal solution

Match with Optimal Solution Matched (✓), Not matched (✗)		Problems						
		1	2	3	4	5	6	
BFS	Proposed Method	MOM	✓	✗	✓	✓	✗	✓
	Existing Method	NWC	✗	✗	✗	✗	✗	✗
		LCM	✗	✗	✗	✗	✗	✓
		VAM	✗	✗	✓	✓	✗	✓

A Statistical (Probability) test has been performed to check whether the proposed algorithm performs better than the other BFS algorithm. Hence, the performance of the proposed method is found to be significantly better than the other method in 67% (4/6) cases are getting an optimal solution.

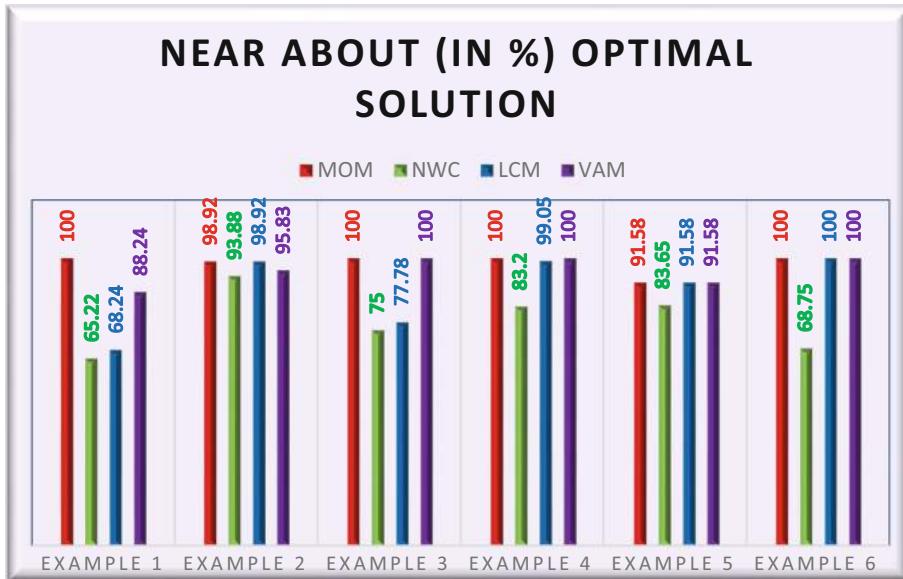


Fig. 1. Result analysis graph of Table 7

Table 9. Probably optimal: proposed method is probably optimal, methods are compared using different types of problems and concluded probably optimal result

Basic feasible solution		Probably optimal
Proposed method	MOM	0.67
Existing method	NWC	0
	LCM	0.17
	VAM	0.5

Summary

This paper introduces a new approach to find a basic feasible solution of transportation problem. The objective is to find a better feasible solution which is near to optimal solution or sometimes exact optimal solution. In order to minimize the transportation cost of product we had to find the product with the minimal per unit transportation cost. This was done by solving the proposed Mount order method. In addition, the optimal result for the selected problem was obtained. Notable, the time complexity remains the same as that of the Vogel's approximation method. It may be observed in Fig. 1 and 2 that the % of similarity is near or equal to the optimal result (from Table 8 and Table 9). The difference between rests of two examples which are not similar to optimal are also very small i.e. $930 - 920 = 10$, $475 - 435 = 40$. Since the proposed method results

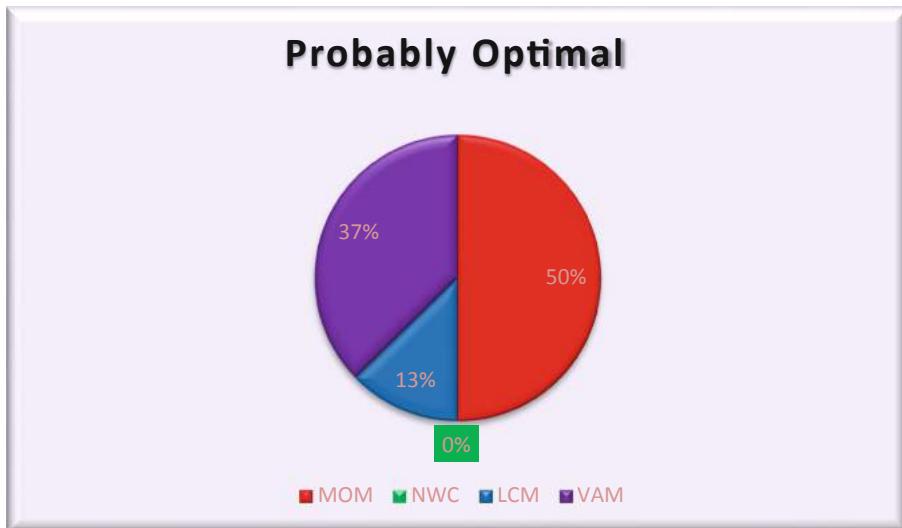


Fig. 2. Result analysis graph of Table 9

are very similar to optimal solution, then it may be implied that the method proposed to find BFS is better than existing method.

6 Conclusion

Operation research (OR) is a scientific study and it offers powerful tools to understand and analyze certain classes of problems to achieve better outcomes. The solution of a transportation problem (TP) has two stages, first to identify the basic feasible solution and next to get the optimal solution (OS). A set of choices that result in the condition being satisfied is called BFS. In this article, a new approach named Mount order method (MOM) for finding a BFS of TP is proposed. There are some popular methods which are used to find a BFS of TP are North-west corner Rule (NWC), Least cost method (LCM), and Vogel's approximation method (VAM). I present and solve a set of six numerical problems by using these three existing methods and compare the obtained result with proposed MOM. The BFS is obtained by using VAM and MOM, and then the result of these BFS is used to find the optimal solution. In some instances, I realized the result obtained by proposed MOM is very close to the optimal solution. The proposed algorithm shows reasonable complexity to find BFS. With the solution provided by MOM can easily reach to the optimal solution in very less number of iterations using the Steppingstone method or Modified distribution method (MODI).

References

1. Romero-Conradoa, A.R., Castro-Bolañoa, L.J., Montoya-Torres, J.R., Jiménez-Barros, M.Á.: Operations research as a decision-making tool in the health sector: a state of the art. *DYNA* **84**(201), 129–137 (2017)

2. Khan, A.R., Vilcu, A., Uddin, Md.S., Ungureanu, F.: A Competent algorithm to find the initial basic feasible solution of cost minimization transportation problem. *Bul. Inst. Politeh. Din Iasi Romania Secția Autom. Calcul. LXI(LXV)* (2), 71–83 (2015)
3. Ehrgott, M., Holder, A.: Operation research methods for optimization in radiation oncology. *J Radiat. Oncol. Inf.* **6**, 1–41 (2014)
4. Islam, M.A., Haque, M., Uddin, M.S.: Extremum difference formula on total opportunity cost: a transportation cost minimization technique. *Prime Univ. J. Multidisc. Quest* **6**, 125–130 (2012)
5. Islam, M.A., Khan, A.R., Uddin, M.S., Malek, M.A.: Determination of basic feasible solution of transportation problem: a new approach. *Jahangirnagar Univ. J. Sci.* **35**, 101–108 (2012)
6. Roberts, J.: Your guide to building a successful strategic plan. [https://www.gartner.com/en/insights/strategic-planning](https://www.gartner.com/en/insights стратегическое планирование)
7. Khan, A.R.: A re-solution of the transportation problem: an algorithmic approach. *Jahangirnagar Univ. J. Sci.* **34**, 49–62 (2011)
8. Khan, A.R., Banerjee, A., Sultana, N., Islam, M.N.: Solution analysis of a transportation problem: a comparative study of different algorithms. *Bull. Polytech. Inst. Iasi Romania Sect. Text. Leathersh.* (2015)
9. Ahmed, M.M., Khan, A.R., Uddin, M.S., Ahmed, F.: A new approach to solve transportation problems. *Open J. Optim.* **5**(1), 22–30 (2016)
10. Ahmed, M.M., Khan, A.R., Ahmed, F., Uddin, M.S.: Incessant allocation method for solving transportation problems. *Am. J. Oper. Res.* **6**(3), 236–244 (2016)
11. Uddin, M., Khan, A.R., Kibria, C.G., Raeva, I.: Improved least cost method to obtain a better IBFS to the transportation problem. *J. Appl. Math. Bioinf.* **6**(2), 1–20 (2016)
12. Thomas, S.J.: Capacity and demand models for radiotherapy treatment machine. *Clin. Oncol.* **15**(6), 353–358 (2003)



Analysis of Improved User Experience When Using AR in Public Spaces

Vladimir Barros^(✉), Eduardo Oliveira, and Luiz Araújo

Cesar School, Recife, Brazil

contato@cesar.school

Abstract. The new immersive technologies have been offering several possibilities and creating new markets around the world. Projects using Augmented Reality offer different experiences in the most diverse segments, including culture. If well designed in public spaces, AR can become a powerful tool in situations that simulate, facilitate, and favour cultural learning, making it more interesting and interactive. This article aims to investigate the improvement of relationship of users with sculptures in public spaces through the use of an AR tool. This article aims to investigate the improvement of relationship of users with sculptures in public spaces through the use of an AR tool. We build on this analysis to identify improvements promoted by the use of AR tool in the learning and engagement process of the users about relevant statues of remarkable personalities of local culture. The idea was to integrate technology, culture, and design so that public spaces, which contain these monuments, could be more visited and have their values re-meant through information.

Keywords: Augmented reality · User centered design · Culture

1 Introduction

Bleicher and Comarella [4] define that immersive virtual environments are the ones that cross the limits of physical space and allow interaction to happen immersed in simulated realities. User engagement is much greater because of this opportunity for interaction created by the endless opportunities for approach. The Augmented and Virtual Reality devices (Fig. 1) provide experiences that mix the physical and virtual worlds through three-dimensional images. Technically, Gnipper [8] says that the process of these immersive environments undergoes a gradation, starting with Augmented Reality, considered much larger than Virtual Reality for the year 2023. Due to its practicality, AR has a certain advantage over VR you need glasses for it to work. Its daily use will generate new experiences that will attract more interested parties, moving to a Mixed Reality (AR plus VR), and, after a familiarization, arrive at Virtual Reality. This way, the user would have time to understand these applications with distinction.

It is generally agreed today that AR is generally related to areas of advertising and entertainment. According to SILVA [19], the American multinational technology PTC presented a research on the marketing market made by AR. Economic movements

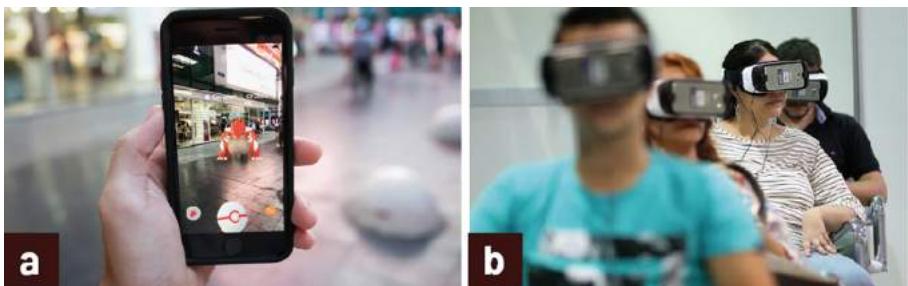


Fig. 1. Augmented realities (a) and Virtual (b) Sebrae; Website 2017.

reached US\$ 7 billion in 2019 and by 2021, this figure is expected to reach US\$ 63 billion, an increase of 800%.

In the cultural area, there are great opportunities for use, a local example is the Circuit of Poetry in Recife, a series of 12 life-size sculptures of great artists of Pernambuco literature and music scattered throughout the neighborhood (Fig. 2).



Fig. 2. Statue of Ascenso Ferreira of the circuit of poetry of Pernambuco. JETRO ROCHA/BORAALI, website 2017.

Some of these statues present necessary structural problems of repairs, such as on the information plates of each of them. These include information about the lives of the honorees, which is lost due to vandalism and wear and tear. The statue of Ascenso Ferreira (Fig. 2) was the basis of study for the strategic location where there is intense pedestrian circulation. This would facilitate the research process with the public, punctuating the beneficial possibilities for the use of AR in the public spaces that the monuments meet.

2 Literature Review

2.1 The User Centered Design

According to LOWDERMILK [13], usability and UCD will change the way people interact with your application, creating a rich experience with solid design principles. This transforms the digital age and viscerally affects all economic and social structures. EVANS and KOEPFLER [7] also define the UCD as an improved man-world experience

process with technology and consider it as a means to engage people for the living environment.

So instead of focusing design on technology, it focuses on how to increase reality with necessary, user-defined things. First, it is necessary to choose the public, monitoring, and evaluating their behavior in the face of technological application. They emphasize that defining the public it is necessary to investigate users in their wild environment, rather than in a created environment. When they engage in objective activities in their habitats, one can analyze how they are affected by AR and know if we are increasing experience and improving it.

The study can bring advantages that minimize costs and foster innovations applied to real life. This research also goes through Norman's design principles [15] that help to understand aspects that transform the creation of products and services:

1. Visibility: It is all that the user can perform in sequence. When the functions are out of sight, the same does not know how to use them.
2. Feedback: Return of information for the user to continue a task and can present itself in a sensory way with audio, tactile, visual, or their combinations. Without it, the user can perform improper actions or repeat them more than once.
3. Restrictions: Restricting the number of choices your artifact becomes safer and easier to use. An objective and unique way make the process more intuitive.
4. Mapping: Where we relate two things, in this case, between virtual environments, the user, and the results of this relationship in the world.
5. Consistency: Similar tasks are obtained with operations and similar elements visually. This transformation of the interface gives greater control to the user.
6. Affordance: An object that allows people to know how to use it by physical obviate and suggestion of interaction.

According to JACOBS [10], the urban space is where interactions between people are generated and where various individual and collective desires are mixed, making it an inseparable element of the city. Likewise, SANTOS and VOGEL [18] argue that the city is formed from the view of its users, having in its daily lives and the variables and complex relations man-space the basis of urban knowledge.

MOREIRA and AMORIM [14] defend projects such as *Archeoguide* (Fig. 3), which recreates architectural monuments of the city of Olympia in Greece, home of the Olympic Games of Antiquity, bringing invaluable to world culture.

The potential of AR in the performance of cultural heritage is seen from documentation, publication, and information dissemination, allowing the user to visualize and interact with objects and historical data. LIMA [12] evaluates that the AR used in cultural scripts brings a freer, contextualized, and complete experience. Moreover, by AZUMA [3], HUGUES [9] AND ZILLES BORBA, ZUFFO, and MESQUITA [20, 22], this article noted RA as a tool for:

- Virtually project images to increase the real-world experience;
- Maintain the sense of presence in the real environment, without the immersion of other tools of simulation of objects or spaces, increasing the use of other users;
- Combination of hybrid worlds, real and virtual, through artifact;



Fig. 3. Temple of Hera in Olympia: (a) Ruins, Present Times; (b) SIMULATION in AR. VLAHAKIS, Website; 2002.

- Complement the real world and not replace the physical environment.

For AZUMA [4], there is an addition in the real environment when using AR for interaction with the virtual, attaching some data entry commands, image capture, and sensory and geolocated devices. This information is processed for assimilation into a peripheral that connects the two environments in real-time. PINHO, CORRÊA, NAKAMURA and JÚNIOR [16] affirm it is necessary to observe the two environments on certain tasks, emphasizing that interactive construction goes through techniques to assertively elaborate an AR project (Table 1).

Within this line, LANTER and ESSINGER [14] argue that intuitive design has products represented by visual models suitable for UX and the activity that was intended. ABRAS, MALONEY-KRICHMAR, and PREECE [1] also cite the use of UCD to build knowledge experiments that connect the real world and people's life experiences. For this, the user needs the exploration control so that the exposed material is understandable and simple in its essence. Plan all the actions of the public aiming at possible errors, it is important to give the same, the chance of recovery if by chance lose the driving of the use of the product.

2.2 Cultural Application

Around culture, Arias [2] reveals the importance of understanding it through its roots. Assimilating these origins makes people more susceptible to understanding the need to keep them alive in memory. According to LÓSSIO and PEREIRA [11], there is a loss of this context throughout life, where culture suffers interference from some factors for stagnation and a lack of appreciation of its popular nature. It is possible to mention the influence of globalization that has come to change the way people behave in the face of popular culture rituals and celebrations. Tradition is valued by custom, but there is no custom of valuing its real meanings. Then, the subject should be discussed under the perspective of cultural sustainability and how to provide pleasure in the segment, so that it becomes a desirable and salable product, in order to generate business and income.

Table 1. Introduction to virtual and augmented reality; 2018. Adapted by the Author; 2019.

Tasks	Real environment	Virtual environment
Manipulation of objects	Object manipulation is usually done with tools or by hand	Tool selection it's complicated
Communication and commands by means of voice	The possibility of communication with other users by means of voice is fundamental importance in the process interactive between users	The technology of voice recognition it's still precarious
Measurement of objects	The possibility of measuring environment objects is quite natural for real applications	It is still difficult and little need the possibility of measuring objects in virtual environments
Annotation of Information about environment objects	Annotating information text and graphics on paper or notice boards is extremely simple and useful in the process interaction in real environments	The entry of texts and numbers is poorly developed in virtual environments

It was justified in this study the reflection and production of effective alternatives with the use of AR to solve problems, examples of the little use of public spaces and the scarcity of cultural information about its monuments.

2.3 Cultural Application

The research was characterized as qualitative when the behaviors of users in the face of an immersive process in AR in the monuments of Recife were analyzed. A device called Digital Circuit of Poetry (Fig. 4) was created, prototypical to present information about



Fig. 4. Identity of the digital poetry circuit project. Created by the author; 2019.

the work of Ascenso Ferreira to the user. Using design methodologies were analyzed, developed, implemented and observed solutions to achieve results.

3 Research Methodology

3.1 Sample

Observing the Global Mobile Consumer Survey 2019 produced by DELOITTE [5], the public that participated in the survey was selected. This definition of the user profile for a first interview was intended to obtain a control group for the application of the tests. With the document, a clipping of the people who started the tests was defined.

Soon after, the Structural Survey Questionnaire was applied to define the control group of future tests. In a semi-structured interview, the user's habits and interests were a little more known.

Taking into account the use of mobile frequently and easily with connected devices, the public was segmented into male and female individuals with an average age between 18 and 36 years (Fig. 5).

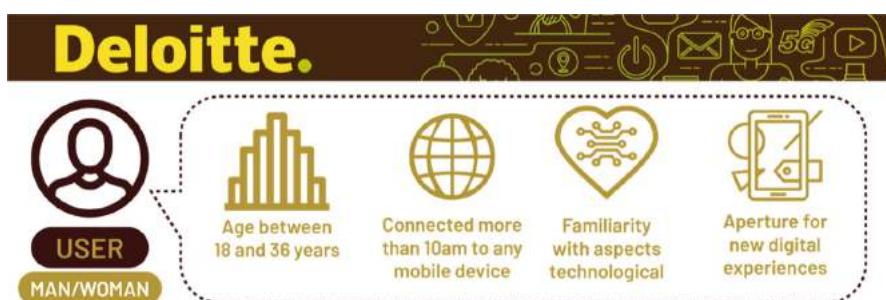


Fig. 5. The user profile is defined for the survey based on DELOITTE data. Prepared by the Author 2019.

Thirty individuals were chosen according to the theory of TULLIS and ALBERT [21], in which they classify the samples under two points: those that identify usability problems in the interaction, with 3 or 4 participants to detect the most significant points, and those that point observations of a process involving many tasks. In the latter it takes much more than four users and claims that coming closer to completion, more people are needed to identify the missing problems. This study of the degree of confidence in the sample (Table 2) defines the success of the research in a number defined in a group of participants who were able to perform the tasks. According to the authors, as it gets closer to the conclusion, more people are needed to identify the missing problems.

Table 2. Example of how confidence intervals change as a function of the sample size. Prepared by the Author; 2020.

Number successful	Number of participants	95% low confidence	95% high confidence
4	5	36%	98%
8	10	48%	95%
16	20	58%	95%
24	30	62%	91%
40	50	67%	89%
80	100	71%	86%

Note: These numbers indicate how many participants in a usability test have successfully completed a given task and the confidence interval for that average completion rate in the larger population

3.2 Research Steps

The general design of the research followed the model of DRESCH, PACHECO, and JÚNIOR [6] for technological studies and developed as follows:

Problem Identification: The use of immersive technologies in public environments was investigated to raise hypotheses that indicated the construction of research thinking. Information was organized to plan the best way to collect the data, in addition to exploring the urban spaces where the little-visited monuments are located. Intrinsic problems were raised for each guide to reach the result, observing convergent points in similar studies that led to an opportunity for further analysis. **Directed readings:** With directed studies, articles, and materials related to the three areas proposed by the problem were researched: RA, UCD, and public spaces.

Identification and proposition of artifacts and configuration of problem classes: At this stage, we sought to identify how AR could improve user interaction with the external environment, and how the production of a device would mean a location before little or not moved. Using an artifact would make the data collection process simpler to observe the possible solutions to the problem.

Artifact design: After the research related to the fields involved and with the support of the directed studies, the Brainstorm technique was used to launch a solution that best suited the problem. In this phase, a dense work of ideation for development was sought, bearing in mind that this process was feasible and scalable.

Artifact development: At this time began to run project processes creation of the artifact: 3D modeling, Ux, and mapping of the monument. The prototyping occurred in phases with test cycles until the material was executed efficiently.

Artifact evaluation: The artifact was evaluated with experts by analyzing the effectiveness of the user experience in a real test field. This moment was crucial for necessary adjustments aiming at an assertive application of the research.

Semi-structured interviews (1st part): As discussed in item 4.1, the control group was defined through applied research on users defined by the DELOITTE survey [5].

Semi-structured interviews (2nd part) and application of tests: For better understanding, this process was divided into four moments:

- Initially, the 2nd part of the semi-structured interview was applied with The Technical Questionnaire of Research 1 (see item 4.2 of this article). In this phase, we tried to understand the relationship of users with monuments directly, focusing on their habits at the time of analog appreciation, without any interference from technology.
- In the second part, the user was presented with the Augmented Reality artifact on the monument of Ascenso Ferreira claiming authorial poetry. The necessary information with all the steps for using the device has been made available for testing via a mobile device.
- In the third part, after following the instructions for the use of AR and having tested all the material in all its possibilities and functions, the Technical Questionnaire of Research 2 was applied (see item 4.3 of this article). Following the Rohrer desirability study [17], a short user experience questionnaire was created to evaluate the use of AR applied in the monument, and some points of User-Centered Design associated with information exposure were observed.
- In the last stage, research technical questionnaire 3 (see item 4.4 of this article) was used to understand how the user sees interactivity, usability, and desirability in a more incisive way. His objective questions concern the use of this type of technology applied to culture. After applying this method (Fig. 6) it was observed that the data obtained with the use of technology can solve the problem proposed by this dissertation.

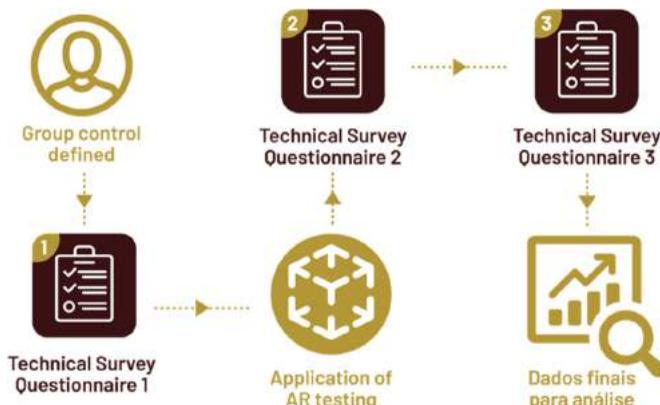


Fig. 6. Method of the 2nd part of semi-structured interviews and application. Prepared by the Author; 2019.

- Explanation of learning: With the results of the tests in hand, the feedbacks were cataloged for the evaluation of possible improvements.

- Conclusions: At this stage, all the results were formalized to generate a final observation with the judgment of the solution and even the new possibilities generated for future studies.
- Generalization for a class of problems: This stage aimed to present the possibility of implementation for similar situations.
- Communication of the results: Presented here all the lessons learned by the process and the possibility of future developments. Table captions should be placed above the tables.

As this research project was proposed in 2019, it is important to note that adaptations have been made in some of the processes on account of the COVID-19 pandemic. The development of the artifact, the application of the interviews, and the application of the tests underwent some changes to comply with the guidelines of social distancing suggested by the World Health Organization (WHO).

All studies and methodologies were idealized for development in a real environment, with the application of tests in the public space. But it was necessary to find an alternative in which users could experience as close to the initial idea within their own homes. Thus, the DSR was adjusted to guide the research and conduct the operation from the beginning, with planning, to the analysis of the final data.

4 Results

4.1 Structural Questionnaire

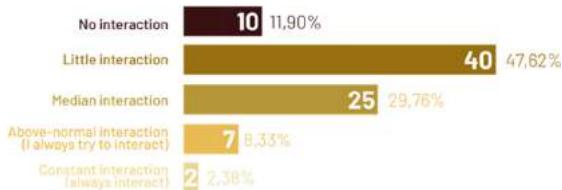
This semi-structured questionnaire aimed to know a little more about the habits and interests of the user to define the control group, selecting 30 interviewees to follow the other stages of the research with the following results (Fig. 7, 8, 9, 10, 11, 12, 13, 14 and 15):

1. What is the system of your cell phone?

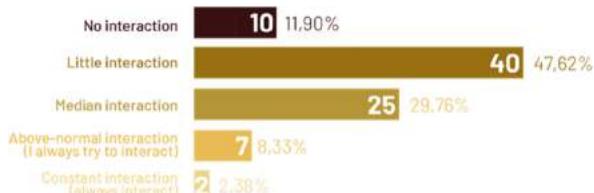


Fig. 7. Structural questionnaire result P1.

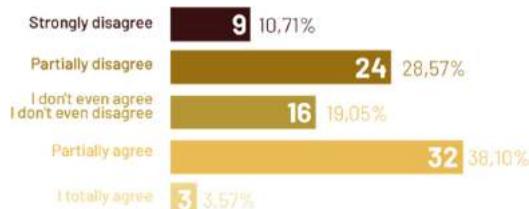
2. What is your level of interaction with architectural works, sculptures and Monuments?

**Fig. 8.** Result structural questionnaire P2.

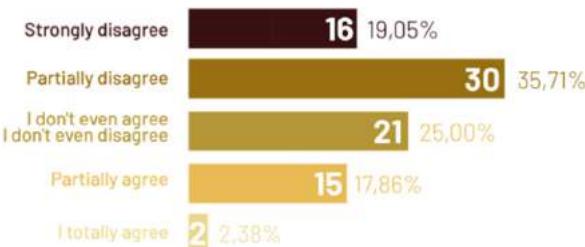
3. Do you consider the information presented about architecture, statues and monuments that are easy to see?

**Fig. 9.** Structural questionnaire result P3

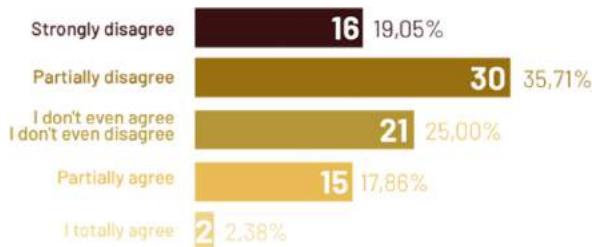
4. Is the information presented educational?

**Fig. 10.** Result structural questionnaire P4.

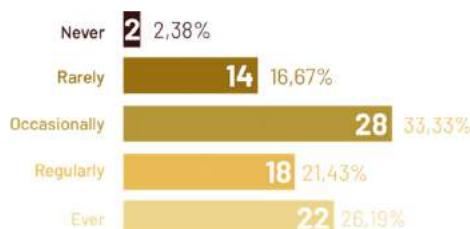
5. Are you satisfied with the information presented?

**Fig. 11.** Result structural questionnaire P5.

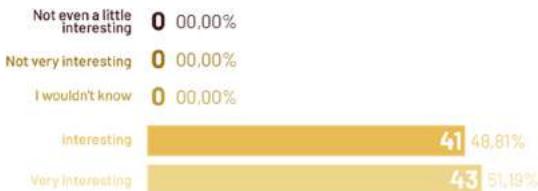
6. How do you get this information? (You can score more than one)

**Fig. 12.** Result structural questionnaire P5.

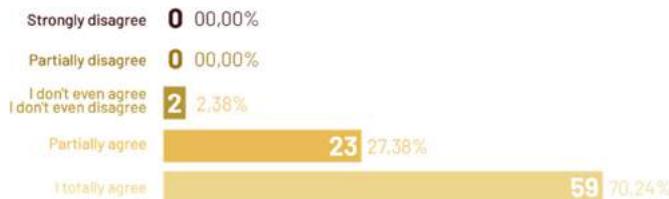
7. How often do you use your mobile phone to obtain tourist information and cultural?

**Fig. 13.** Result structural questionnaire P7.

8. What would you think of some application with interactive audio and video applied to these sculptures?

**Fig. 14.** Result structural questionnaire P8.

9. About the use of smartphones do you think it would facilitate the process?

**Fig. 15.** Result structural questionnaire P9.

4.2 Technical Questionnaire 1

This first semi-structured technical interview aimed to understand the relationship of the control group with the monuments directly, focusing on their habits at the time of analog appreciation, without any interference from technology. We obtained the following results (Fig. 16, 17, 18, 19 and 20):

1. What is your profession?

30 people with different works like: designers, engineers, journalists, and students.

2. What is your age group?

**Fig. 16.** Result technical questionnaire-1 P2.

3. About monuments and museums: How much time do you spend observing the monuments in the streets or museums?



Fig. 17. Result Technical Questionnaire-1 P3.

4. How often do you see innovative modes of presentation of Monuments?

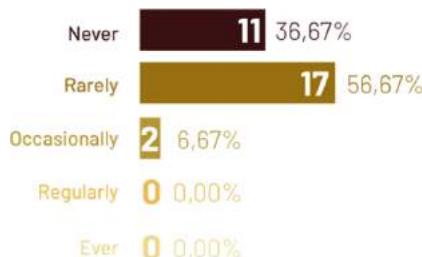


Fig. 18. Result technical questionnaire-1 P4.

5. My interest in the monuments has a direct connection with the way where they are presented.

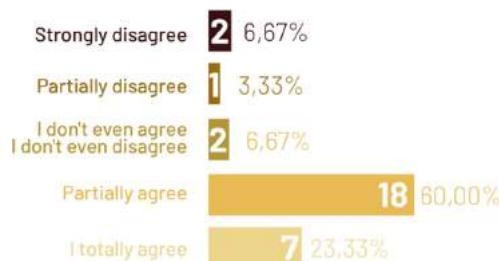


Fig. 19. Result technical questionnaire-1 P5.

6. You would again enjoy exhibits or monuments already seen previously?

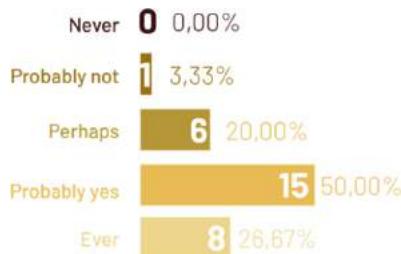


Fig. 20. Result technical questionnaire-1 P6.

4.3 Technical Questionnaire 2

After following the instructions for the use of the Augmented Reality application and testing application (Fig. 4), this second technical questionnaire was answered. With it was expected to analyze the UX regarding the use of AR applied in the monument, observing some points of the UCD associated with the exposure of information. According to the experience in the use of the AR of the Digital Circuit of Poetry, the scale below was used to measure the user's experience by marking the closest to the adjective identified during the proposed test. We obtained the following results (Fig. 21, 22, 23, 24, 25, 26, 27 and 28):

1. What is your view on the use of AR in the proposed test (fluidity level)?

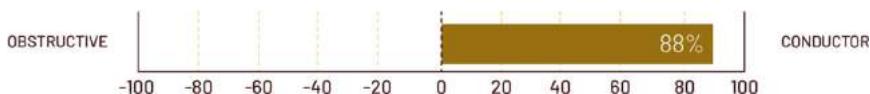


Fig. 21. Result technical questionnaire-2 P1.

2. What is your view on the use of AR in the proposed test (difficulty level)?

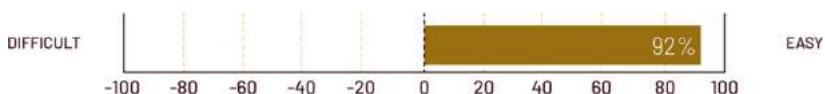


Fig. 22. Technical questionnaire-2 P2 Result.

3. What is your view on the use of AR in the proposed test (efficiency level)?

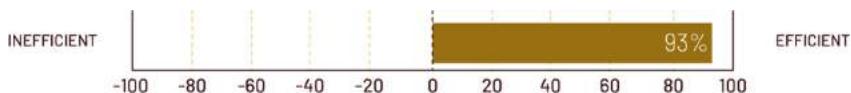


Fig. 23. Result technical questionnaire-2 P3.

4. What is your view on the use of AR in the proposed test (level of comprehensibility)?

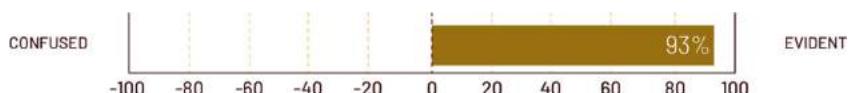


Fig. 24. Result technical questionnaire-2 P4.

5. What is your view on the use of AR in the proposed test (level of dynamism)?

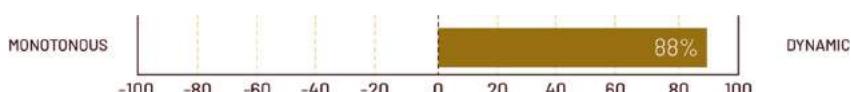


Fig. 25. Result technical questionnaire-2 P5.

6. What is your view on the use of AR in the proposed test (level of interest)?

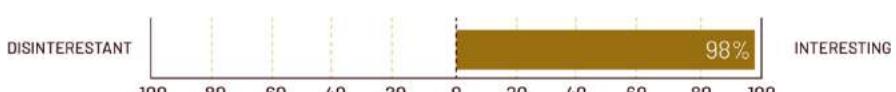


Fig. 26. Result technical questionnaire-2 P6.

7. What is your view on the use of AR in the proposed test (level of innovation)?

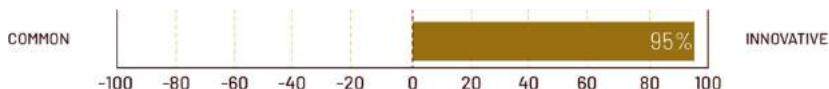


Fig. 27. Result technical questionnaire-2 P7.

8. What is your view on the use of AR in the proposed test (temporal level)?

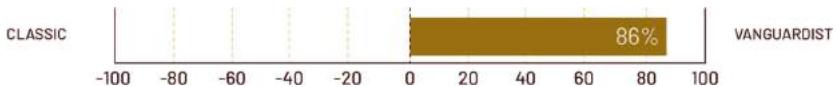


Fig. 28. Result technical questionnaire-2 P8.

4.4 Technical Questionnaire 3

This final step aimed to know the user's view more directly with the interactivity, usability, and desirability of the artifact. Through semi-structured research, the experience with AR was related to objective questions that alluded to the use of this type of technology applied to culture. We obtained the following results (Fig. 29, 30, 31, 32, 33, 34, 35 and 36):

1. How interactive did you find the AR application in this context?

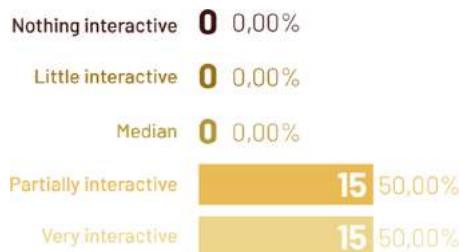


Fig. 29. Technical questionnaire-3 P1 result.

2. The interactivity presented by the application aroused my interest about the appreciation of the monument.



Fig. 30. Result technical questionnaire-3 P2.

3. How do you see the use of the application?



Fig. 31. Technical questionnaire-3 P3 result.

4. The possibility of pausing and continuing (on the button) with the recital of poetry made the experience easier to use.

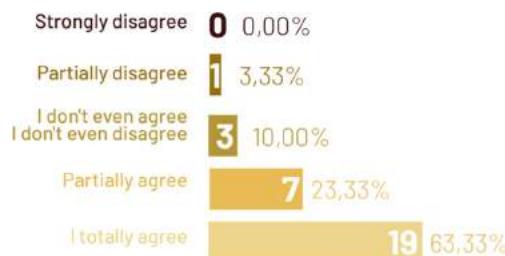
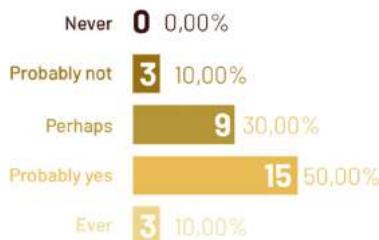


Fig. 32. Result technical questionnaire-3 P4.

5. The presentation of the monument in AR, reciting a poem, brought a desire over the history of the work.

**Fig. 33.** Result technical questionnaire-3 P5.

6. Would you pay for this type of product in culture-oriented AR?

**Fig. 34.** Result technical questionnaire-3 P6.

7. Would you indicate this type of cultural product in AR?

**Fig. 35.** Result technical questionnaire-3 P7.

8. Why would you consume these cultural products? (You can score more than one)

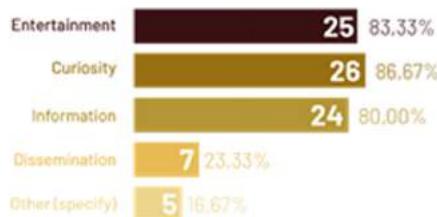


Fig. 36. Result technical questionnaire-3 P8.

5 Recommendations

In its entirety, the numbers were very positive when we observed the application's support in a group that consumes technology daily. This fact alone is already a positive indicator for the defense of cultural proposals involving technology and public spaces. The structural questionnaire provided basic information about the cultural positioning of the public:

They interact little with the city's monuments;
Time or time again they find it difficult to visualize the information of the works;
They observe partial gains in the educational aspect of the information provided;
They feel some dissatisfaction with the amount of information presented;
They use their mobile phone to facilitate the retrieval of information;
They felt they would have a greater interest if the works were more interactive.

In technical questionnaire 1, already more immersive in AR, users generally opined on the following questions:

- They do not easily find innovative applications in monuments.
- Much of it has direct interest when monuments have a differentiated presentation.
- Much of it would visit an exhibition again.

With technical questionnaire 2, there are important technical considerations for the assertiveness of the research, since users consider positive AR in all aspects and following the table of TULLIS and ALBERT [21] (Table 2), we have 91% confidence, since all were able to complete the action defined in the test. They considered the application of fluid AR during the process, easy to use, efficient for its destination, evident in the content presented, dynamic in the interaction presented, and with a high level of innovation.

Finally, technical questionnaire 3 presented an objective explanation of the interviewees, being more evaluative about this application in the public space; considers the application interactive.

They were interested in enjoying the monument.
He sees that the application would facilitate immersion.
It brought desire about the place and monument.
Most would pay for consumption and indicate this type of product.

6 Conclusion and Future Research

The use of Augmented Reality as a hard-hitting way for the occupation of public space is analyzed throughout the process. It is expected to resignify these places within the city through a more immersive experience, addressing the cultural context over a new look. Accompanying this process also gives the possibility of expansion to other monuments within the Neighborhood of Recife, since they have several other cultural architectural works.

The users approached showed great willingness to consume cultural products in their State, however, the way they present themselves does not give the possibility of a search and greater immersion in the subject. The artifact visibly caused a user's inquiry into the life of the poet honored in the Poetry Circuit. This fact itself already makes it notorious how an immersive approach can generate knowledge and interconnect information that goes unnoticed. Even the recommendation of the users tested was the inclusion of some functionalities to make the artifact more interactive: a menu that gave access to an audiobook where they could permeate other poetry; telling of causes about the poet's life, where he would speak of his life story; and a set of questions and answers about the life of the statue honoree. These possibilities show how positive the interaction with the user was and that they are really in need of stimuli to consume products that usually do not go through this type of approach.

It is important to highlight the possibility of scaling this project to other works in Recife since the capital has several other works that are not within the Poetry Circuit and go through the same problem of abandonment of public space. This digitization of history would make the most visited and prepared environments for the reception of local or outside visitors. This strengthening brings again in the knowledge of human history; after all, the identity of a people is intrinsically linked to their culture.

It is expected that the object of study Circuit da Digital da Poesia can be part of local digital tourism, presenting the culture of Pernambuco in a more didactic and desirable way, always instructing on the roots of its ancestors. This time, not only for consumption at home, as was the case of the test, more in the place where the statues meet. Only then can we make public what is ours: the knowledge about our traditions and the place where they were born.

References

1. Abras, C., Maloney-Krichmar, D., Preece, J.: User-centered design. In: Bainbridge, W., et al.: Berkshire Encyclopedia of Human-Computer Interaction. <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.94.381&rep=rep1&type=pdf>. Accessed 14 May 2019
2. Arias, P.G.: La cultura. estrategias conceptuales para comprender a identidad, la diversidad, la alteridad y la diferencia. LNCS. https://digitalrepository.unm.edu/cgi/viewcontent.cgi?article=1009&context=abya_yala. Accessed 13 July 2020
3. Azuma, R., Baillot, Y., Behringer, R., Feiner, S., Julier, S., Macintyre, B.: Recent advances in augmented reality. IEEE Comput. Graph. https://www.researchgate.net/publication/3208983_Recent_advances_in_augmented_reality_IEEE_Comput_Graphics_App. Accessed 26 Mar 2020
4. Comarella, R.L., Bleicher, S.: Experimentação de recursos didáticos. <https://moodle.ead.ifsc.edu.br/mod/book/view.php?id=82391&chapterid=16198che>. Accessed 25 Oct 2020

5. DELOITTE GLOBAL: A mobilidade no dia a dia do brasileiro. <https://www2.deloitte.com/br/pt/pages/technology-media-and-telecommunications/articles/mobile-survey.html>. Accessed 22 July 2020
6. Dresch, A., Pacheco, D., Antunes Júnior, J.A.V.: Design Science Research: método de pesquisa para avanço da ciência e tecnologia. 1st edn. Bookman, Porto Alegre (2015)
7. Evans, K., Koepfle, J.: The UX of AR: toward a human-centered definition of augmented reality. <http://uxpamagazine.org/the-ux-of-ar/>. Accessed 15 July 2020
8. Gnipper, P.: Especialista faz 5 previsões sobre o futuro das realidades virtual e aumentada. <https://canaltech.com.br/rv-ra/especialista-faz-5-previsoes-sobre-o-futuro-das-realidades-virtual-e-aumentada-119024/>. Accessed 20 Sept 2020
9. Hughes, O., Fuchs, P., Nannipieri, O.: New augmented reality taxonomy: technologies and features of augmented environment. <https://hal.archives-ouvertes.fr/hal-00595204/document>. Accessed 24 Aug 2020
10. Jacobs, J.: Morte e vida das grandes cidades. 3 nd edn. WMF, Martins Fontes, São Paulo (2014)
11. Lanter, D., Essinger, R.: User-centered design. In: Richardson, D., et al. (eds.) International Encyclopedia of Geography: People, the Earth, Environment and Technology. <https://onlinelibrary.wiley.com/doi/full/10.1002/9781118786352.wbieg0432>. Accessed 21 May 2018
12. Lima, Ma.C.G.V.: Realidade Aumentada Móvel e Património no Espaço Público/Urbano. Dissertação de Mestrado em Gestão e Programação do Património Cultural, apresentada à Faculdade de Letras da Universidade de Coimbra. <https://eg.uc.pt/bitstream/10316/36094/1/Realidade%20Aumentada%20Movel%20e%20Patrimonio.pdf>. Accessed 25 Nov 2019
13. Lowdermilk, T.: Design Centrado no Usuário. Novatec Editora. São Paulo (2013)
14. de Souza Moreira, L.C., de Amorim, A.L.: Realidade Aumentada e patrimônio cultural: apresentação, tecnologias e aplicações. https://www.researchgate.net/publication/275153985_REALIDADE_AUMENTADA_E_PATRIMONIO_CULTURAL_APRESENTACAO_TECNOLOGIAS_E_APlicACOES Accessed 29 Sept 2019
15. Norman, D.: The Design of Everyday Things. Basic Books, New York (1988)
16. Pinho, M., Corrêa, C., Nakamura, R., Júnior, J.: Introdução a Realidade Virtual e Aumentada. Editora SBC, Porto Alegre (2018)
17. Rohrer, C.P.: Desirability studies: measuring aesthetic response to visual designs. <https://www.xdstrategy.com/desirability-studies/>. Accessed 15 July 2020
18. dos Santos, C.N.F., Vogel, A.: Quando a rua vira casa: A apropriação de espaços de uso coletivo em um centro de bairro. 3 nd edn. Projeto, Rio de Janeiro (1985)
19. Silva, B.: Marketing mais que real. <https://www.istoeinheiro.com.br/marketing-mais-que-real/>. Accessed 13 Apr 2020
20. Tori, R., da Silva Hounsell, M. (eds.): Introdução a Realidade Virtual e Aumentada. Editora SBC. Porto Alegre (2018)
21. Tullis, T., Albert, B.: Measuring the User Experience Collecting, Analyzing, and Presenting Usability Metrics. Morgan Kaufmann Publisher, New York (2013)
22. Zilles Borba, E., Zuffo, M., Mesquita, F.: Uma nova camada na realidade: realidade aumentada, electrónica e publicidade. http://www.lasics.uminho.pt/ojs/index.php/cecs_ebooks/article/view/2897. Accessed 23 Aug 2020



Autonomous Vehicle Decision Making and Urban Infrastructure Optimization

George Mudrak^(✉) and Sudhanshu Kumar Semwal

Department of Computer Science, University of Colorado at Colorado Springs, Colorado Springs, CO 80918, USA

Abstract. Highly mobile populations can quickly overwhelm an existing urban infrastructure as large numbers of people move into the city. The urban road networks having been constructed for less traffic will quickly become congested leading to diffusion of traffic and a greater spread of congestion as secondary roads are increasingly utilized. Further unwanted and potentially fatal consequences such as increased accident, stress, driver anger, and road rage will increase. Traditional methods of addressing the growth in traffic density would include increasing lane count and construction of new roads, and can be costly. Yet traditional responses do not take into account the growing number of connected vehicles and soon, possible autonomous vehicles will interact with human-driven cars. In considering a driving population of autonomous vehicles, we identify an increased range of possible traffic management strategies for collaborative experiences. The vehicles themselves can operate according to their own goals and the urban infrastructure can also be enhanced to a greater degree of self-management. This paper will explore the ideas, techniques and approach behind creating an Agent Based Modelling environment to support the interaction of autonomous vehicles with a smarter urban infrastructure. Procedural content generation (PCG) and agent based modelling concepts will be applied in establishing the modelling framework.

Keywords: Agent based modeling · Autonomous vehicles · Smart infrastructure · Smart cities · Procedural generated cities · Population noise maps

1 Introduction

Cities manage their infrastructures based on projections regarding future increases or decreases of the population over time. When a population rapidly increases for any reason, infrastructures become strained. In particular, rapid population growth with corresponding mobility increases traffic congestion within these urban areas [1]. Congestion does not remain localized and isolated but rather flows along and disperses from influential road segments [2] and important road intersections, identifying of which is difficult [3]. Congestion itself remains only part of the concern as increased traffic, stress, anger, fear and anxiety directly could impact driver and pedestrian safety [4, 5]; at times for the worse, with fatal accidents. To address these and other concerns, we evaluate urban road infrastructure and to utilize the best tools possible. Future tools will quickly be added

to the set for urban planners as autonomous vehicles (AVs) continue making their way into everyday use. Thus increasing influence of autonomous vehicle as the ubiquitous car continues making inroads.

Many researchers are already contributing to the study of AVs [6]. The automotive and IT industries [7] along with government support [8] have made the goal for self-driving cars their research priority. Between 2014 and 2017, more than \$80 billion was spent on the AV field [9] and states such as Nevada, USA have authorized state wide testing of AVs [10] along with a number of cities across the North America and Europe [11].

So what defines autonomous or self-driving vehicles? AVs can drive themselves from one location to another location in “autopilot” mode, i.e. no driver operational inputs, using various in-vehicle technologies and sensors [12] along with various decision making software models. They operate on the road in the same fashion as people operated vehicles. Immense contribution is expected in the area of aging population mobility and others.

AVs must make ongoing decisions regarding their inputs and goals, in order to allow them to operate in real-world traffic. Obeying existing traffic laws is at the foundation of what these vehicles must enforce.

For autonomous vehicles to gain acceptance into society, considerations must include more than vehicle dynamics and legal adherence. The passenger’s priorities should be included. These priorities would certainly include travel time to answer the question of reaching the destination in a timely or even early manner. Was the trip comfortable for the passenger or did the vehicle behave in such a manner as to cause some physical, mental or emotional discomfort?

These vehicles should be able to operate and adjust in a world with common safety and traffic laws along with substantial cultural variance in adherence to the letter of the law. Imagine taking a 4902 mile road trip with your vehicle from Berlin, Germany to Delhi, India. While the core traffic laws remain consistent driving through southern Europe, across Turkey, Iran, Afghanistan and Pakistan, the local population’s adherence to laws, patience, tolerance for deviations, etc. contains considerable variance. The following paper discusses the Agent-Based AVs implementations along with a review of existing research.

2 Background and Literature Review

2.1 Agent Based Modeling of Complex Systems

Complex Systems represents a growing domain of computing sciences in which any number of components, systems, agents, etc. interact to represent a system, solve a problem, or even explore an idea [14]. Complex systems are modeled using various techniques, though the use of agent based modeling continues finding favor within a number of fields [15]. While each field has created their own nomenclature and characteristics for representation within complex systems, they have in common that they require autonomous agents and often represent some aspect of human-based interaction [13, 14, 16].

ABMs provide an ideal solution to modeling specific types of theory and implementation for a number of reasons such as establishing a normalized environment for repeatability, especially with the involvement of multiple disciplines [17], economic impacts, and/or ethical concerns requires a participant safe approach [13]. ABMs lend themselves quite readily towards multi-domain modeling. From a multi-disciplinary view ABM work takes into account environmental factors as well as complex human behaviors, examples being the development of predictive models relating to burglary [18], population impacts of processes of re-urbanization within metropolitan areas [19] and work in simulating transportation [69]. Economic centric ABMs explore the various costs associated with retail and service oriented markets along with population impacts [20–22]. Work in especially sensitive areas that possess high levels of ethical and social concerns benefit substantially from the use of ABMs. Such work would include individual bullying behaviors [23] and group based bullying behaviors [24]. Models have been presented in which they theories regarding the link between urban sprawl and income segregation [25]. Additional work illustrates the use of ABMs for exploration of city planning and population dynamic zones [65]. Further, city and population ABMs have even been used within the gaming industry for example the well-known SimCity series [66] and even the current Cities Skylines [83] PlayStation 4 game.

The use of agent based models requires the combination of concepts and tools from a variety of fields such as computing sciences, sociology, psychology, economics, game theory, physics, biology, and medicine. The ABM contributions to both the research and industry fields are made across multiple disciplines and well established.

Complex Systems

For our purposes, Complex Systems, adheres to the general views espoused by Mitchell [14] and Gilbert's book [13]. In brief, Mitchell [14] identifies three commonly exhibited properties of complex systems complex collective behavior, signaling and information processing, and adaptation. She further provides the following insight into the basic elements comprising a complex system as illustrated in Table 1:

Table 1. Elements of a complex system

Environment	The space in which the actors operate, inclusive of its own set of properties
Actors	Atomic individuals that inhabit the environment and possess their own set of properties
Behaviors	Actor behaviors to interact with other actors and the environment
Population	A collection of homogeneous agents

Taken as is, Mitchell's statements of the attributes of complex systems provide a framework in which agents, and hence agent based modeling, can operate.

Gilbert brings us into the ABM space by elaborating on the elements found and mapped up to the general framework provided by Mitchell [13]. Gilbert's complimentary view can be seen in Table 2:

Table 2. Complementary view for the elements of a complex system

Ontological correspondence	A direct correspondence between the computational agents in the model and real-world actors
Heterogeneous agents	A mix of dissimilar agents operating according to their defined preferences and rules
Environment	The environment in which the agents operate
Interaction	The ability for agents to affect one another
Bounded rationality	Limiting the cognitive abilities of the agent and thus the degree to which they can optimize their utility
Learning	Agents learn from their experiences through evolutionary and social learning

An outcome of ABMs lies with ontological correspondence covering emergence of non-trivial, non-coded behaviors resulting from the interaction of fundamental decisions by the autonomous agents as they grow, adapt, learn and interact with their environment and one-another.

An additional benefit of Complex System ABMs is found in Epstein and Axtell's [16] critique of homogeneous agents highlighting Gilbert's [13] specific "heterogeneous" criteria. In their work, Social Science from the Bottom Up", Epstein and Axtell present that the use of rational homogeneous agents, removes the richness and unpredictability from a population. While this modeling reaffirms macro-level behaviors, it does not emphasize micro-level behaviors and as such represents a barrier to one of the hallmarks of complex systems - emergent behavior, [16].

As evident with our discussion from previous sections, the application of ABMs fits extremely well with modeling Autonomous Vehicles (AVs) given consideration for the multi-disciplinary nature of urban road infrastructure, vehicles, and human behavior. It is also readily apparent given the economic, market and even ethical/social impacts that reaffirm the use of ABMs in this domain. Real-world physical research includes such as areas as pedestrian activity [26], lane management [27], or any other driver assisting technologies; these are not included in this study as they are outside the scope of our research at this time.

As in real-life, our ABM's vehicle agents were able to determine a route based on a starting location and a destination to determine a possible route at a reasonable cost. Kohout and Erol [28] demonstrated two key points applicable to our approach in their work on agent-based vehicle routing. The first being that the use of a stochastic approach (which is preferred in ABMs) to routing yielded a viable alternative to a centralized routing heuristic, which is usually avoided in favor of local interactions; the second being the routing occurring in a changing environment. This is significant in that an ABM utilizes stochastic behavioral responses based on environmental factors representing "real-time" traffic information that vehicular agents use to model driving successfully [29]. This interdependence at every tick of the simulation has the capacity to exhibit complex and multi-dimensional behavior. Further viewing of the routing for a vehicle on the urban road system, demonstrates that in fact a connected graph exists in

which every intersection or dead-end represents a node while all the roads themselves represent the edges. Applying this understanding of the known shortest path algorithms ensures that our vehicles can travel from start to destination [30].

Further work has illustrated the impact that a change in population can have upon an urban environment [19] and reinforces the importance of being able to account for population growth or decline. Especially in consideration that no city remains an enclosed static entity but rather undergoes dynamic organic changes over time.

ABM Modeling Tools

A large number of ABM modelling frameworks exist supporting both single and multi-domains such as teaching, visualization, financial markets, manufacturing, business, healthcare, social sciences, technologies, design, etc. [67, 68]. Focusing on the human element of modeling emphasizes a framework in the social sciences domain and with our need to include custom code requires a framework that supports plug-ins for non-native algorithms and code. Additionally, utilizing a framework where an active user community exists provides options for support and example code. Further, it was critical that the framework would support large numbers of differentiated, autonomous actors with a relatively reasonable learning curve for the utilized language.

From prior experience and evaluation, it was known that the ABM framework Net-Logo [64] could meet the necessary environment and interaction needs. We are quite familiar with the framework and programing language having utilized it for prior ABM research and simulations [15, 23, 24, 65].

2.2 Behavioral Influencers

The modern vehicle operator today must contend with driving distractions as a common component of everyday driving [31] and the potential for those distractions to negatively impact their performance [32, 33]. For our purposes, we examine two sources for the distractions and behavioral impactors that affect driving performance and decision making.

Cultural Influencers Using ABMs

Driving in small-town America traffic does not compare to driving in New Delhi or New York City traffic, for a number of reasons – even driving between large cities such as New Delhi and New York City. It is not difficult to project the impact of common and recognized in-vehicle distractions [34] given an obvious increase in vehicles. Culture represents an additional factor taken into account as an influencer on driving behavior using ABMs. Even if driving left or right hand side, still rules seems to be same, following similar traffic law encouraging similar flows. While the vast majority of countries present the same basic traffic laws, driver compliance, deviation tolerance and behavioral responses can vary substantially across the world's countries. Research focusing on cultural differences have utilized the driver behavior questionnaire (DBQ) and have confirmed that cultural behavioral responses are in fact country/culture dependent with an understandable impact on the driving experience [35]. Further work has shown that

while behavioral responses vary, driver anger in response to perceived affronts remains consistent across cultural boundaries [36, 37].

Occupant Influencers

As discussed, the environment contains numerous distractions and other influencers that affect a driver's ability to safely operate a vehicle and can lead to stress manifesting itself in aggressive driving [38]. It has been shown that aggressive driving influenced by anger leads to traffic accidents [39] as angry drivers would be slower in responding to various traffic events [40].

Anecdotally, it is known that unchecked anger is not what people typically exhibit. Researchers include a form of "forgiveness" factor in their research [41]. While the effectiveness of de-escalation strategies such as reminders of past individual driver's road-rage meets with success [42] with varying effectiveness across cultures [43], it warrants inclusion for us.

Adjusting our view of anger, and giving it the general term impatience, allows us to examine the behavioral impact in a non-emotionally biased fashion. Combined with our understanding that both cultural and personal responses from drivers influence their performance and decision making, these types of factors are accounted for within the model. This allows us to address autonomous vehicle behaviors themselves in similar fashion to how drivers view situations so that "progress" is made. Thus we recognize autonomous vehicles themselves must have a concept of impatience in relation to meeting a primary goal of delivering passengers on time to their destinations. As such, for example, an autonomous vehicle cannot allow itself to be stonewalled by pedestrian traffic, or stay on a predetermined route no matter the traffic situation.

2.3 Urban Planning

Approaches to urban planning often take several factors into account when optimizing the road infrastructure. This includes city, state, country and region influenced. For example, in the central and mountain portions of the United States, substantial road infrastructure exists given the distributed city planning and number of vehicles. While in contrast, other parts of the world prioritize pedestrian traffic [44, 45] possibly reducing the number of available roads along with other strategies for pedestrian safety [46].

Research into the domain of artificial city creation is ongoing and typically focused on the film and game industries [47, 66]. These cities are often created via procedural content generation (PCG) so that a consistent set of constraints and criteria are applied for a viable result [48]. By using the PCG approach and varying input factors, we can create a new-infinite amount of content that adheres (cities) to the same construction and layout constraints [49].

We applied the principles behind PCG as a means of generating a stochastic and distributed starting population as our structure and creating the road network as our function of the structure [50].

Population Density Visualization

Our primary concerns regarding population generation are two-fold. First, we desired a PCG approach so that populations can be created quickly and repeatedly [50] and

secondly, a nonlinear agglomeration population distribution as that corresponds to real world cities [51, 52].

Foundational work has been done to meet these goals in other areas such as procedural image creation and effects [53] by Ken Perlin in an effort to create less “machine-like” computer graphics [54] through the creation of an algorithm now referred to as Perlin Noise and an enhanced version by K. Perlin referred to as Simplex Noise. In both cases, the algorithm procedurally generates nonlinear “noise” exhibiting elements of agglomeration.

One means of approaching creation of a population data set focuses around working with non-targeted data sets related to some portion of a population [55, 56]. In these cases a variety of information feeds into what becomes an aggregated data set. For us, the limitations of this approach include the need for access to a number of data sources that we do not have and what essentially would become an emphasis on creating the “right” data set. We additionally evaluated a more recent approach to noise generation termed “blue noise”. In the blue noise approach by Mitchell, Ebeida, Awad, et. al. [57] they illustrate an algorithm to generate noise distributions with good quality and performance across different dimensions. Their approach while being PCG friendly, yields uniformly distributed noise maps that do not exhibit our need for agglomeration. The noise is too uniform to correspond with a real-world varied population density distribution, and therefore we did not utilize this approach [57].

We found an additional approach in using two-dimensional Perlin noise and used it to implement the city generation [58]. The Simplex noise algorithm generates a nonlinear matrix of smoothed and agglomerated values, we can see that it meets our criteria as also being a PCG friendly algorithm. As such, we have chosen to make use of this approach in generating our working populations.

Infrastructure Visualization

All road networks exist to move members of a population from point-A to point-B and for our road generation, we defined and met similar core requirements as we established with our population generation. First, we desired a PCG based approach to ensure that the process could be quick and repeatable. Secondly we found and agreed that a heuristic based approach performed well in road network creation [59].

Road networks can be duplicated or even procedurally generated over existing maps with extracted semantics [60] or even built up from image-derived templates or raster maps [61, 62]. Knowing that we have a population density map provided, we use Simplex noise to focus on the procedural and heuristic aspects for our roads.

We found Parish and Muller’s approach to road generation quite insightful especially in its application of global and local rules to “craft” the road network [59]. Application of these rules, or viewed another way “constraints” facilitated a flow to the generated road networks resulting in a greater correspondence with real road infrastructures.

As a whole, for our urban planning, we recognized the need for a base population to exist as the foundation necessary for the generation of an urban road network. We identified various research that highlighted the ability to procedurally generate a population and from that data, a road network. Being able to achieve this as PCG and repeatedly, provides substantial benefit as we can focus our exploration on the AV and passenger behaviors

involved without the distractions around creating the perfect infrastructure. We further increase our ability to test our behaviors against stochastically created environments.

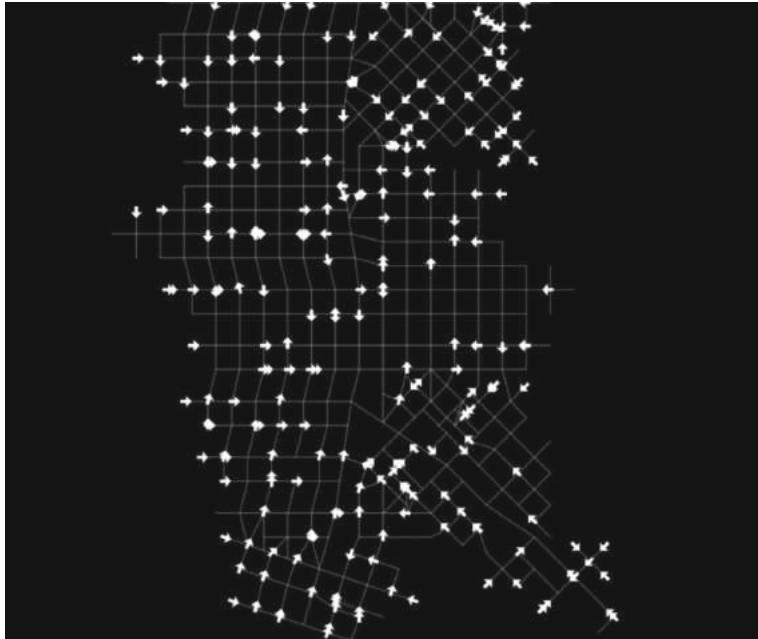


Fig. 1. Road infrastructure with vehicles present

Vehicles

NetLogo [63, 64] supports agent and link visualization as illustrated in Fig. 1. These visuals are enabled through use of a standard library of off-the-shelf icons and the ability to create/edit custom icons and links [84]. This support enables vehicles to be represented graphically in customizable fashion with respect to both the icon itself and the color. Further links, the visual relationship between agents, can be customized allowing for a variety of display options applicable to our usage of links as road segments. This graphical capacity within NetLogo provides all we need for visualization of vehicles and road segments within our ABM framework.

Vehicle Route Planning

Vehicle route planning within an urban road infrastructure is comparable to traversing an undirected connected graph in which all vertices or intersections are linked by one or more roads [85]. Essentially any path finding algorithm that would guarantee a result could apply [70–72]. For our purposes, any well-known and established algorithm that met our needs for finding a guaranteed route and is well-known would suffice. We

selected the well-known A* algorithm as it met our needs to deliver a guaranteed route, is well established and a baseline for other research [71].

Toll Planning

The topic of road pricing has been a consideration of economic and transportation research for years, [73] with works performed to determine acceptability [74]. Recent research demonstrates that use of road pricing can reduce traffic jams [75] with the additional benefit of reducing vehicle emissions [76].

Behavioral research has been conducted on what changes drivers will make and the acceptance of those changes [77] along with economic impact in terms of costs, [75] or through positive incentives [77]. We elected to exclude these driver behavioral factors from consideration while accepting the foundational position that road pricing via tolls does affect traffic. There, we elected to include toll support within our framework.

Positional Currency

The use of tokens to change or incentivize traveler behavior has been occurring for some time [81] with recent research folding game theory into use and commuter behavior [82]. These tokens represent another approach to the concept of road pricing that can be applied per traveler. We rebranded the general “token” and called it “commuter token”.

Having a commuter token represents one-half of the solution for AVs. The other half can be found in the models behind game theory negotiation. These have been examined as far back as 1961 [78] with newer research into robotic resource negotiation [79] and multi-agent enterprise collaboration [80].

Through use of the commuter tokens, we empower the AVs with a means to offer their own incentives encouraging cooperation throughout the AVs lifetime [77].

The ability for AVs to self-negotiate position through use of a commuter token or commuter currency allows to greater traffic flow.

3 Conclusion

As the automotive domain continues its unstoppable drive towards truly autonomous vehicles we have identified how proving support for a smarter urban infrastructure can in combination with AV data, optimize traffic flow. Further in this survey paper we have presented the ideas that AVs must operate in more than just their primary culture and thus an exploration of disparate cultural influences must take place. We also present visualization of AVs on the PCG generated road-map using NetLogo™. We feel that through agent based modelling we can optimally explore the diverse operational values of both the autonomous vehicle and a smart urban infrastructure. This research would yield information applicable to a required autonomous vehicle transitional period as they approach saturation and also how a city could respond during and after the adoption phases.

References

1. Liu, C., Vu, H.L., Leckie, C., Anway, T.: Spatial partitioning of large urban road networks. In: Proceedings of EDBT (2014)

2. Liu, C., Vu, H.L., Islam, S., Anwar, T.: RaodRank: traffic diffussion and influence estimation in dynamic urban road networks. In: CIKM, Melbourne, pp. 1671–1674 (2015)
3. Page, L., Brin, S.: The anatomy of a large-scale hypertextual web search engine. *Comput. Netwo. ISDN Syst.* **30**(1–7), 107–117 (1998)
4. Schmidt-Daffy, M.: Fear and anxiety while driving: differential impact of task demands, speed and motivation. *Transp. Res. F: Traffic Psychol. Behav.* **16**, 14–28 (2012)
5. Groeger, J.A., Stephens, A.N.: Do emotional appraisals of traffic situations influence driver behaviour? In: Behavioural Research in Road Safety: Sixteenth Seminar, pp. 49–62 (2006)
6. 2025AD The Automated Driving Community. <https://www.2025ad.com/latest/top-universities-for-autonomous-driving/>
7. CBInsights. <https://www.cbinsights.com/research/autonomous-driverless-vehicles-corporations-list/>
8. ABC News. <https://abcnews.go.com/US/companies-working-driverless-car-technology/story?id=53872985>
9. Brookings. <https://www.brookings.edu/research/gauging-investment-in-self-driving-cars/>. Accessed 10 Oct 2020
10. Las Vegas SUN. <https://lasvegassun.com/news/2017/jun/20/autonomous-vehicles-in-nevada-roll-forward-with-ne/>
11. Showbiz CheatSheet. <https://www.cheatsheet.com/money-career/cities-have-driverless-cars.html/>
12. Gartner IT Glossary. <https://www.gartner.com/it-glossary/autonomous-vehicles/>. Accessed 10 Oct 2020
13. Gilbert, N.: Agent-Based Models, Series: Quantitative Applications in the Social Sciences. SAGE Publications (2008)
14. Mitchell, M.: Complexity a Guided Tour, 1st edn. University Press, Oxford (2009)
15. Mudrak, G.: Modeling aggression & bullying: a complex systems approach. Master Thesis UCCS (2013)
16. Axtell, R., Epstein, J.M.: Growing Artificial Societies - Social Science from the Bottom Up. Brookings Institution Press (1996)
17. Axelrod, R.: Agent-based modeling as a bridge between disciplinse. *Handb. Comput. Econ.* **2**, 1565–1584 (2006)
18. Evans, A., Jenkins, T., Malleson, N.: An Agent-based model of burglary. *Environ. Plann. B: Urban Anal. City Sci.* **36**(6), 1103–1123 (2009)
19. Rauh, J., Rid, W., Hager, K.: Agent-based modeling of traffic behavior in growing metropolitan areas. *Transp. Res. Proc.* **10**, 306–315 (2015)
20. Moreno, A., Nealon, J.: Agent-based applications in health care. In: Applications of Software Agent Technology in the Health Care Domain, pp. 3–18. Birkhäuser, Basel (2003)
21. Colson, A., Chisholm, D., Dua, T., Nandi, A., Laxminarayan, R., Megiddo, I.: Health and economic benefits of public financing of epilepsy treatment in India: an agent-based simulation model. *Epilepsia* **57**(3), 464–474 (2016)
22. Evans, A.J., Birkin, M.H., Heppenstall, A.J.: Genetic algorithm optimisation of an agent-based model for simulating a retail market. *Environ. Plann. B: Urban Anal. City Sci.* **34**(6), 1051–1070 (2007)
23. Mudrak, G., Semwal, S.: Modeling aggression and bullying: a complex systems approach. In: Annual Review of CyberTherapy and Telemedicine, pp. 187–191. IOS Press (2015)
24. Mudrak, G., Semwal, S.: Group aggression and bullying through complex systems agent based modeling. *Ann. Rev. Cyberther. Telemed.* **14**, 189–194 (2016)
25. Buchmann, C.M., Schwarz, N., Guo, C.: Linking urban sprawl and income segregation - findings from a stylized agent-based model. *Environ. Plann. B: Urban Anal. City Sci.* **46**(3), 469–489 (2019)

26. Batty, M.: Agent-based pedestrian modeling. *Environ. Plann. B. Plann. Des.* **28**(3), 321–326 (2001)
27. Tewolde, G., Zhang, X., Kwon, J., Dang, L.: Reduced resolution lane detection algorithm. In: IEEE AFRICON, pp. 1459–1465, November 2017
28. Erol, K., Kohout, R.: In-time agent-based vehicle routing with a stochastic improvement heuristic. In: AAAI, pp. 864–869 (1999)
29. Dia, H.: An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transp. Res. Part C: Emerg. Technol.* **10**(5–6), 331–349 (2002)
30. Rothkrantz, L.: Dynamic routing using maximal road capacity. In: CompSysTech 2015 Proceedings of the 16th International Conference on Computer Systems and Technologies, pp. 30–37 (2015)
31. Feaganes, J., et al.: Driver's exposure to distractions in their natural driving environment. *Accid. Anal. Prev.* **37**(6), 1093–1101 (2005)
32. Feaganes, J., Rodzman, E., Hamlett, C., Reinfurt, D., Stuffs, J.: The causes and consequences of distraction in everyday driving. *Assoc. Adv. Automot. Med.* **47**, 235–251 (2003)
33. Feaganes, J., et al.: Distractions in everyday driving (2003)
34. Regan, M., Young, K.: Driver distraction: a review of the literature. NSW: Australasian College of Road Safety, pp. 379–405 (2007)
35. Ozkan, T., Lajunen, T., Tzamalouka, G., Warner, H.W.: Cross-cultural comparison of drivers' tendency to commit different aberrant driving behaviours. *Transp. Res. F: Traffic Psychol. Behav.* **14**(5), 390–399 (2011)
36. Gras, M.E., Cunil, M., Planes, M., Font-Mayolas, S., Sullman, J.M.: Driving anger in Spain. *Pers. Individ. Differ.* **42**(4), 701–713 (2007)
37. Yao, X., Jiang, L., Li, Y., Li, F.: Driving anger in China: psychometric properties of the driving anger scale (DAS) and its relationship with aggressive driving. *Pers. Individ. Differ.* **68**, 130–135 (2014)
38. Shinar, D.: Aggressive driving: the contribution of the drivers and the situation. *Transp. Res. F: Traffic Psychol. Behav.* **1**(2), 137–160 (1998)
39. Kontogiannis, T.: Patterns of driver stress and coping strategies in a Greek sample and their relationship to aberrant behaviors and traffic accidents. *Accid. Anal. Prev.* **38**(5), 913–924 (2006)
40. Trawley, S.L., Madigan, R., Groeger, J.A., Stephens, A.N.: Drivers display anger-congruent attention to potential traffic hazards. *Appl. Cogn. Psychol.* **27**(2), 178–189 (2012)
41. Dahlen, E.R., Moore, M.: Forgiveness and consideration of future consequences in aggressive driving. *Accid. Anal. Prev.* **40**(5), 1661–1666 (2008)
42. Takaku, S.: Reducting road rage: an application of the dissonance-attribution model of interpersonal forgiveness. *J. Appl. Soc. Psychol.* **36**(10), 2362–2378 (2006)
43. Roskova, E., Lajunen, T., Kovacsova, N.: Forgivingness, anger, and hostility in aggressive driving. *Accid. Anal. Prev.* **62**, 303–308 (2014)
44. Fatian, H., Gota, S., Mejia, A., Leather, J.: Walkability and pedestrian facilities in asian cities. In: ADB Sustainable Development Working Paper Series, vol. 17, February 2011
45. Pinet-Peralta, L.M., Short, J.R.: No accident: traffic and pedestrians in the modern city. *Mobilities* **5**(1), 41–59 (2010)
46. Roe, M., Shin, H., Viola, R.: New York City pedestrian safety study & action plan. New York City Department of Transportation, New York City (2010)
47. Kelly, G., McCabe, H., Whelan, G.: Roll your own city. In: DIMEA 2008 Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts, pp. 534–535 (2008)
48. Meijer, S., Velden, J.V., Iosup, A., Hendrikx, M.: Procedural content generation for games: a survey. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **9**(1), Article No. 1 (2013)

49. Smith, G.: Understanding procedural content generation: a design-centric analysis of the role of PCG in games. In: CHI 2014 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 917–926 (2014)
50. Kavak, H., Crooks, A., Kim, J.: Procedural city generation beyond game development. *SIGSPATIAL Spec.* **10**(2), 34–41 (2018)
51. Lobo, J., Strumsky, D., West, G.B., Bettencourt, L.: Urban scaling and its deviations: revealing the structure of wealth, innovation and crime across cities. *PLoS ONE* **5**, e13541 (2010)
52. Karmeshu, P.K., Sikdar, P.K.: On population growth of cities in a region: a stochastic nonlinear model. *Environ. Plann. A: Econ. Space* **14**(5), 585–590 (1982)
53. Perlin, K.: An image synthesizer. *SIGGRAPH Comput. Graph* **19**, 287–296 (1985)
54. Wikipedia Perlin Noise. https://en.wikipedia.org/wiki/Perlin_noise
55. Toint, P.L., Barthelemy, J.: Synthetic population generation without a sample. *Transp. Sci.* **47**(2), 131–294 (2013)
56. Ferreira, J., Jr., Zhu, Y.: Synthetic population generation at disaggregated spatial scales for land use and transportation microsimulation. *Transp. Res. Rec.: J. Transp. Res. Board* **2429**(1), 168–177 (2014)
57. Ebeida, M.S., et al.: Spoke-darts for high-dimensional blue-noise sampling. *ACM Trans. Graph. (TOG)* **37**(2), Article No. 22 (2018)
58. Wijgerse, S.: Generating realistic city boundaries using two-dimensional Perlin noise. Doctoral thesis (2007)
59. Muller, P., Parish, Y.: Procedural modeling of cities. In: SIGGRAPH 2001 Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 301–308 (2001)
60. Bidarra, R., Teng, E.: A semantic approach to patch-based procedural generation of urban road networks. In: GDG 2017 Proceedings of the 12th International Conference on the Foundations of Digital Games, Article No. 71 (2017)
61. Yu, X., Bacui, G., Green, M., Sun, J.: Template-based generation of road networks for virtual city modelling. In: VRST 2002 Proceedings of the ACM Symposium on Virtual Reality Software and Technology, pp. 33–40 (2002)
62. Knoblock, C.A., Chiang, Y.: Automatic extraction of road intersection position, connectivity, and orientations from raster maps. In: GIS 2008 Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Article No. 22 (2008)
63. The Center for Connected Learning and Computer-Based Modeling. <http://ccl.northwestern.edu/Uri.shtml>
64. NetLogo (2019). <https://ccl.northwestern.edu/netlogo/index.shtml>
65. Mudrak, G., Semwal, S.: AgentCity: an agent-based modeling approach to city planning and population dynamics. In: International Conference on Collaboration Technologies and Systems, CTS 2012, pp. 91–96 (2012)
66. Wilson, J.: The Official SimCity Planning Commission Handbook. Osborne McGraw-Hill (1994)
67. Abar, S., Theodoropoulos, G.K., Lemarinier, P., O'Hare, G.M.P.: agent based modelling and simulation tools: a review of the state-of-art software. *Comput. Sci. Rev.* **24**, 13–33 (2017)
68. Kravari, K., Bassiliades, N.: A Survey of Agent Platforms. *J. Artif. Soc. Soc. Simul.* **18**(1) (2015). <http://jasss.soc.surrey.ac.uk/18/1/11.html>
69. Bazzan, A., Klugl, F.: Multi-agent Systems for Traffic Transportation Engineering. IGI Global (2009)
70. Imai, H., Asano, T.: Efficient algorithms for geometric graph search problems. *SIAM J. Comput.* **15**(2), 478–494 (2006)
71. Zhou, R., Hansen, E.A.: Sparse-memory graph search. In: InIJCAI - International Joint Conference on Artificial Intelligence, 9 August 2003, pp. 1259–1268 (2003)

72. Hansen, E.A., Zilberstein, S.: LAO*: a heuristic search algorithm that finds solutions with loops. *Artif. Intell.* **129**(1), 35–62 (2001)
73. Morrison, S.: A survey of road pricing. *Transp. Res. Part A: Gener.* **20**(2), 87–97 (1986)
74. Jaensirisak, S., Wardman, M., May, A.D.: Explaining variations in public acceptability of road pricing schemes. *J. Transp. Econ. Policy (JTEP)* **39**(2), 127–154 (2005)
75. Martin, L.A., Thornton, S.: City-wide trial shows how road use charges can reduce traffic jams. *The Conversation*, November 2017. <https://theconversation.com/city-wide-trial-shows-how-road-use-charges-can-reduce-traffic-jams-86324>
76. Bigazzi, A.Y., et. Al.: Road pricing most effective in reducing vehicle emissions. *Phys.Org.* (2017). <https://phys.org/news/2017-10-road-pricing-effective-vehicle-emissions.html>
77. Ettema, D., Knockaert, J., Verhoef, E.: Using incentives as traffic management tool: empirical results of the ‘peak avoidance’ experiment. *Int. J. Transp. Res.* **2**(1), 39–51 (2010)
78. Kuhn, H.W.: Game theory and models of negotiation. *J. Conflict Resolut.* **6**(1), 1–4 (1962)
79. Cui, R., Guo, J., Gao, B.: Game theory-based negotiation for multiple robots task allocation. *Robot* **31**(6), 923–934 (2013)
80. Trappey, A.J., Trappey, C.V., Ni, W.C.: A multi-agent collaborative maintenance platform applying game theory negotiation strategies. *J. Intell. Manuf.* **24**(3), 613–623 (2013)
81. Kearney, A., De Young, R.: Changing commuter travel behavior: employer-initiated strategies. *J. Environ. Syst.* **24**(4), 373–393 (1996)
82. Kracheel, M., McCall, R., Koenig, V.: Studying commuter behaviour for gamifying mobility (2014)
83. Cities Skylines, Sony Playstation, June 2020. <https://www.playstation.com/en-us/games/cities-skylines-ps4/>
84. NetLogo Shapes Editor. <https://ccl.northwestern.edu/netlogo/docs/shapes.html>
85. Wikipedia. Connected Graph. [https://en.wikipedia.org/wiki/Connectivity_\(graph_theory\)#Connected_graph](https://en.wikipedia.org/wiki/Connectivity_(graph_theory)#Connected_graph)



Prediction of Road Congestion Through Application of Neural Networks and Correlative Algorithm to V2V Communication (NN-CA-V2V)

Mahmoud Zaki Iskandarani^(✉)

Al-Ahliyya Amman University, Amman 19328, Jordan
m.iskandarani@ammanu.edu.jo

Abstract. Implementation of simulated Vehicle-to-Vehicle (V2V) communication results using Basic Safety Message (BSM) of 320 Bytes size for the purpose of congestion management and control is carried out. The obtained results used as input to a neural network architecture. The Neural structure uses Weigend Weight Elimination Algorithm (WWEA) to achieve learning in order to predict congestion events by predicting hops count, average network lifetime, average route length and provide them as inputs to the Congestion Management Algorithms (CMA) that correlate between them in order to detect congestion events. This will also aid road traffic designers in their effort for a less congestive roads, and work towards smart applications. Testing results showed that congestion can be determined and predicted in advance, such that a control mechanism can be initiated to re-route traffic with traffic lights timing also intelligently adjusted to alleviate expected traffic congestion and related problems.

Keywords: Intelligent Transportation Systems · Connected vehicles · Congestion · Network lifetime · V2V · Neural networks

1 Introduction

Vehicular Ad hoc Networks (VANETs) are important Networks with the required communication tolerance and delay becoming a critical issue as more safety applications are becoming available and part of the integrated safety strategy of the Intelligent Transportation System (ITS) paradigm. Vehicles in the connected vehicles concept and application would share safety critical information as they communicate with both Infrastructure (V2I) and with other vehicles (V2V).

Road traffic safety and traffic management applications are the two main applications associated with vehicular networks, with interrelation between them, as accidents and road incidents, which are safety issues can also affect smooth traffic flow and needs management strategies [1–5]. Safety applications are used to minimized traffic accidents and consequential injuries and fatalities. Various accidents are road related and some

of them are caused by ill managed roads, particularly at intersections, where, head-on, front and rear collisions can occur.

Vehicular network applications for connected vehicles provide information to vehicles regarding issues related to distance, speed, and location among others. Thus trying to avoid accidents that will lead to congestion. At congestion, vehicles speed and location are affected and thus both safety applications and traffic control application under the appropriate algorithms will provide vital information regarding congestion, with traffic management applications dealing with traffic flow issues to avoid congestion under different conditions. The management and control of traffic operates in both spatial and temporal dimensions, with concentration on speed control and vehicle navigation [6–10].

Dedicated Short Range Communication (DSRC) is considered a viable option for wireless V2V communication. It is important to be able to quantify and assess communication links between connected vehicles in order to optimize communication and enable better traffic control and management. Thus it is of prime importance to examine route or path length connecting two vehicles with its associated parameter of Network Lifetime (NTLT), which is associated with energy consumed in the communication process and reflects communication link usage in terms of time (Path or Route Length) and duration of contact between the considered vehicles. The importance of link sustainability on the network performance is considered an essential metric in routing Basic Safety Messages (BSMs) among connected vehicles and very important in multi-hop routes or paths. Communication link properties is closely related to vehicles speed, physical distance, and route hops, energy consumed, communication time, among others. This indicates that a change in any of these parameters can be used as a sign of traffic irregularity, such as congestion [11–15].

V2V communication using BSM enables vehicles to exchange data covering main parameters such as, speed, location, position, direction, thus enabling decision making through data processing via vehicles On Board Units (OBUs). The exchanged information among vehicles, which covers status like traffic ahead and distance between vehicles among others can be collected either directly form the vehicles (Vehicular Clouds) or through Vehicle-to-Infrastructure communication employing Road Side Units (RSUs) [16–21].

Neural Networks (NN) form an adaptive system that carry out changes to its structure and response behavior as it learns during training process. In its essence, neural network adopts the mechanisms and dynamic structure of biological neurons and process information in a similar manner to the nervous system. Neural Networks learns instead of being programmed, a neural network learns by example (input data) and has the ability after training and inter-neurons connection optimization to generalize and associate inputs with outputs [22–25].

Normally, congestion related models are based on standard models, and are not known for accuracy or optimization. Hence, neural networks can be an efficient and viable tool to optimize existing models and to provide results for complex cases that cannot be formulated easily. Neural networks structures can be trained to predict data under different environments. Neural networks are able to model nonlinear systems using multi-layer neurons with ability for synaptic weights modification as needed to optimize the performance of the network and obtain good convergence [26–28].

In this paper, a neural network model is used to predict congestion on roads. The model uses Weigend Weight Elimination Algorithm (WWEA) [29] and multi-layer architecture which is trained using simulated data resulting from V2V communication using BSM messaging. The simulation is based on actual road structure. The resulting data is communicated to traffic control center and to vehicles to re-route traffic and provide alternative roads, hence, avoid congestion before it actually occurs.

2 Methodology

The main objective of this work is to enable control and avoidance of roads congestion as a function of traffic density depending on V2V wireless communication using BSM messages of 320 Bytes size, using applications which incorporates the following parameters:

1. Hop Count
2. Route Length
3. Communication Time (Hop Time)
4. Network Lifetime (Consumed Energy)

The intention is to provide a managing algorithm interfaced to a neural networks algorithm that acts on the data. The predictions provided by the neural networks algorithms are to be used both in real time management and for design purposes of smart areas. Figure 1 present the Neural Networks architecture used to feed the Congestion Management Algorithm (CMA).

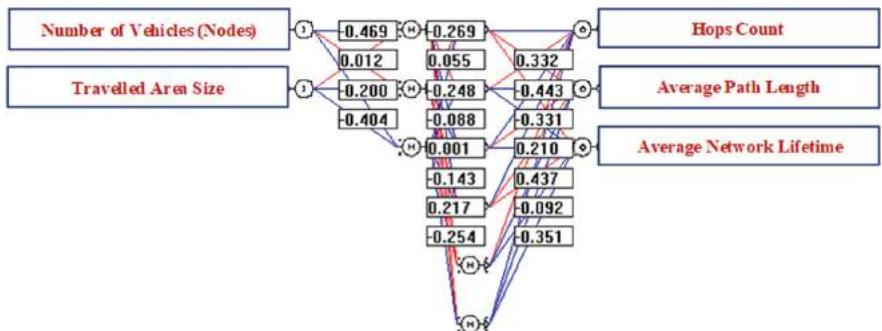


Fig. 1. Neural networks structure for congestion control

The Neural Networks is trained using Weigend Elimination Algorithm (WWEA) as shown in Fig. 2.

WWEA focuses on weights pruning, which will assist in uncovering of dominant feature (s) and contributes towards neural structure fine tuning, hence, optimized results. The WWEA operates on the principles of weight decay with error minimization through an error function. Such optimization process is carried out by adding an overhead term to the original error function employed by the algorithm.

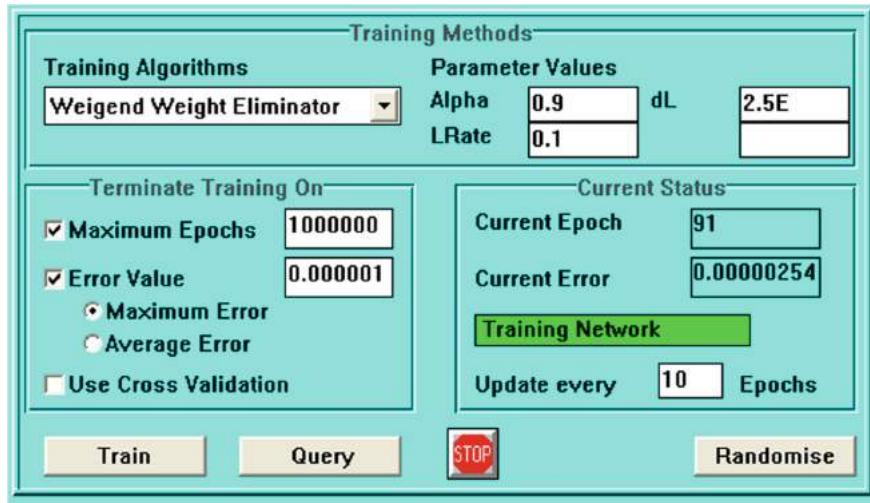


Fig. 2. Neural networks training interface

Based on the overhead, large weights are more affected as they can contribute negatively towards network convergence depending on their position in the network structure. Weights with large values between the input layer and the hidden layer can result in possible discontinuities in the output values resulting in a rough function mapping, while weights between the hidden and output layers can result in instability and oscillations of mapped output function. Thus, WSEA is an efficient algorithm in optimizing and pruning the network structure through its vital overhead function.

The mathematical representation of the application of the overhead function through an error function (EWWE) is described by Eq. (1).

$$E_{WWE} = E_{start} + E_{overhead} \quad (1)$$

The error function (E_{start}) is given by Eq. (2), with the optimizing part ($E_{Overhead}$) is given by Eq. (3).

$$E_{start} = \frac{1}{2} \sum_j (r_j - a_j)^2 \quad (2)$$

$$E_{overhead} = Alpha * \left(\sum_{ij} \frac{\left(\frac{w_{ij}}{w_n}\right)^2}{1 + \left(\frac{w_{ij}}{w_n}\right)^2} \right) \quad (3)$$

Where;

$Alpha$: Weight reduction operator.

w_{ij} : Individual weights of the neural network model.

w_n : Conditioning parameter computed by the WSEA.

r_i : Required Output.

a_i : Actual Output.

The dynamic weight change is calculated through a modified version of the gradient descent algorithm as shown in Eq. (4).

$$\Delta w_{ij} = \left(-LRate * \frac{\partial E_{start}}{\partial w_{ij}} \right) - \left(Alpha * \frac{\partial E_{overhead}}{\partial w_{ij}} \right) \quad (4)$$

Where;

$LRate$: Network learning rate (0 to 1).

The conditioning parameter w_n operates such that the smallest weight is selected from the last iteration or epoch and can operate on a whole set of epochs. The idea is to push the weights values to zero, hence weight elimination. The weight reduction operator.

(Alpha) covers the network complexity aspects as a function of performance for datasets under consideration. The adjustable operator behavior is described by Eq. (5).

$$Alpha = Alpha_0 * \exp(-\lambda(1 - Q_p)) \quad (5)$$

Where;

$Alpha_0$: Scaling operator and can be set to 1.

λ : Multiplication constant,

Q_p : Ratio of correctly classified patterns to the total number of patterns.

The design change in the learning rate ($LRate$) can be also related to the network overall error through Eq. (6):

$$d(LRate) = \gamma * E_{WWE} \quad (6)$$

Where γ is a scaling factor, and should be at least twice the error of the network.

Equation (5) describes a dynamic relationship between $Alpha$ and Q_p , such that an increase in Q_p will results in an increase in $Alpha$, thus increasing the influence of $E_{overhead}$, which pushes smaller weights to zero values (convergence and optimal performance). On the other hand, reduction in Q_p will have the opposite effect, which will adversely affect the state of the network and will disable optimum performance and convergence. The choice of Alpha is critical for a proper pruning process to occur and not to eliminate weights too early, thus loosing accuracy and affecting prediction capability. Thus an indirect connection between Alpha and network error function is established. Also, Eq. (6), applies control of the learning rate ($LRate$) in correspondence to the network error. Both terms in Eq. (5) and Eq. (6) are tied up and correlated through Eq. (3) for an optimum network convergence.

Weigend Weight Elimination Algorithm (WWEA) is a bidirectional algorithm, which is critical for neural network optimization and design.

3 Results and Discussion

Tables 1, 2 and 3 present the initial simulated results for road traffic and V2V communication.

Table 1. 200 vehicles (nodes).

Route no.	Route length	Hops per route	Network lifetime
1	67.1	3	782
2	50.7	2	1270
3	50.6	2	1275
4	50.7	2	1274
5	50.8	2	1269
6	51.0	2	1260
7	51.3	2	1247
8	51.7	2	1230
9	52.2	2	1210
10	52.9	2	1186
11	59.5	3	966
12	55.0	2	1109
13	55.5	2	1091
14	56.1	2	1072
15	56.7	2	1052
16	68.4	2	754
17	70.3	3	718
18	64.9	3	829
19	66.4	3	797
20	61.7	3	906
21	62.1	3	897
22	62.9	3	877
23	64.1	3	849
24	65.6	3	814
25	67.3	3	776
26	69.4	3	736
27	71.6	3	694

(continued)

Table 1. (*continued*)

Route no.	Route length	Hops per route	Network lifetime
28	74.0	3	653
29	72.3	3	682
30	79.0	3	579
31	80.9	3	554
32	82.9	3	530
33	83.1	3	528
34	85.0	3	505
35	95.4	4	407
36	95.5	4	406
37	95.8	4	404
38	96.3	4	400
39	96.9	4	395
40	97.8	4	388
41	98.7	4	381
42	106.6	4	329
43	101.0	4	365
44	99.8	4	374
45	101.4	4	362
46	103.2	4	350
47	105.8	4	334
48	107.8	4	322
49	109.8	4	311
50	111.9	4	300

Table 2. 520 vehicles (nodes).

Route no.	Route length	Hops per route	Network lifetime
1	47.1	2	1435
2	46.9	2	1448
3	46.6	2	1460

(continued)

Table 2. (*continued*)

Route no.	Route length	Hops per route	Network lifetime
4	46.4	2	1470
5	46.3	2	1480
6	46.1	2	1488
7	46.0	2	1496
8	45.8	2	1503
9	43.5	2	1635
10	44.4	2	1583
11	45.4	2	1525
12	46.5	2	1465
13	47.8	2	1403
14	49.1	2	1341
15	50.5	2	1279
16	46.3	2	1475
17	46.2	2	1480
18	46.3	2	1475
19	46.6	2	1460
20	47.1	2	1435
21	47.8	2	1403
22	48.6	2	1364
23	49.6	2	1321
24	50.7	2	1274
25	51.9	2	1225
26	53.2	2	1175
27	54.5	2	1124
28	56.0	2	1075
29	57.3	2	1033
30	58.7	2	989
31	69.4	3	735
32	81.7	3	544
33	82.3	3	537
34	82.2	3	538

(continued)

Table 2. (*continued*)

Route no.	Route length	Hops per route	Network lifetime
35	82.2	3	539
36	82.2	3	538
37	82.4	3	535
38	82.7	3	532
39	81.1	3	552
40	82.5	3	535
41	82.2	3	539
42	83.6	3	522
43	82.2	4	539
44	83.8	4	519
45	88.7	4	467
46	86.6	4	488
47	98.8	4	381
48	100.4	4	369
49	102.1	4	357
50	103.9	4	346
51	105.7	4	334
52	107.7	4	323
53	110.8	4	306
54	112.8	4	295
55	106.1	4	332
56	108.2	4	320
57	110.4	4	308
58	110.0	4	310
59	111.8	4	300
60	110.6	4	307
61	116.6	4	277

Table 3. 600 vehicles (nodes).

Route no.	Route length	Hops per route	Network lifetime
1	52.5	2	1200
2	51.2	2	1249
3	50.1	2	1298
4	50.6	2	1276
5	49.2	2	1339
6	47.8	2	1403
7	46.5	2	1468
8	45.3	2	1531
9	44.2	2	1592
10	43.2	2	1648
11	42.4	2	1699
12	41.8	2	1741
13	41.3	2	1773
14	41.0	2	1792
15	40.9	2	1799
16	41.0	2	1792
17	41.3	2	1773
18	41.8	2	1741
19	42.4	2	1699
20	43.2	2	1648
21	44.2	2	1592
22	45.3	2	1531
23	46.5	2	1468
24	47.8	2	1403
25	49.2	2	1339
26	50.6	2	1276
27	44.2	2	1592
28	44.9	2	1555
29	45.7	2	1510
30	46.6	2	1460
31	47.7	2	1406
32	48.9	2	1349

(continued)

Table 3. (*continued*)

Route no.	Route length	Hops per route	Network lifetime
33	50.2	2	1292
34	51.6	2	1235
35	53.1	2	1178
36	54.0	2	1144
37	55.2	2	1100
38	56.6	2	1055
39	58.0	2	1011
40	74.7	3	642
41	76.6	3	613
42	64.9	3	828
43	66.5	3	793
44	68.2	3	758
45	69.9	3	725
46	71.7	3	693
47	73.5	3	662
48	75.3	3	633
49	77.2	3	605
50	79.1	3	579
51	79.2	3	577
52	81.2	3	551
53	83.3	3	525
54	88.0	3	474
55	104.7	4	341
56	101.6	4	361
57	102.8	4	353
58	104.1	4	345
59	105.5	4	336
60	106.9	4	327
61	101.2	4	364
62	103.2	4	350
63	105.2	4	338
64	107.2	4	326

(continued)

Table 3. (continued)

Route no.	Route length	Hops per route	Network lifetime
65	107.6	4	323
66	108.9	4	316
67	110.3	4	308
68	111.7	4	301
69	109.6	4	312
70	111.5	4	302

Figure 3, 4 and 5 present simulated and fitted curves describing the relationship between route length and V2V pattern of communication over time. From the plots it is realized that the vehicle under consideration forms an incrementing exponential relationship as it travels through blocks of infrastructure. Such an exponential curve fitting covers the overall path the vehicle travels relative to other vehicles, at the same time it provides a mathematically reasonable illustration to the effect of traffic density on increasing the number of hops per route within the same infrastructure, which is non-linear and agrees with real time scenarios. The general fitting curve prescribing the relationship between V2V communication over route length variation and sequential route connectivity (distance and time dependent) is approximated by Eq. (7)

$$V2V(N)_{RL} = \eta * \exp(\beta * SRC) \quad (7)$$

Where;

SRC: Sequential Route Connectivity.

N: Number of Vehicles (Nodes) in a spatial domain.

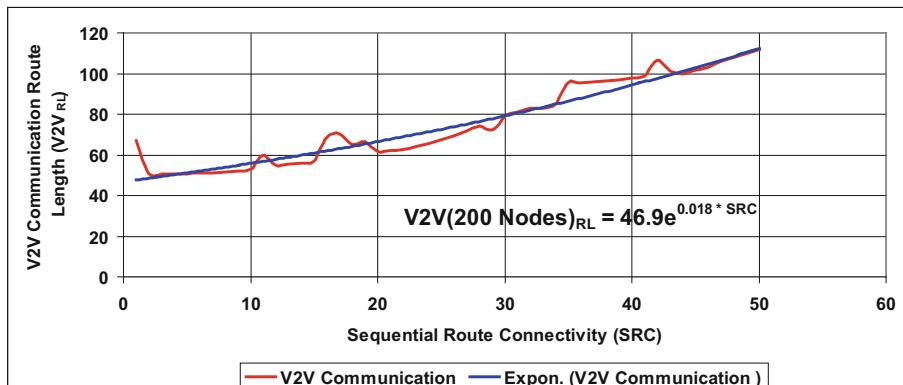


Fig. 3. Relationship between route length and connectivity over time for traffic density of 200 vehicles (nodes)

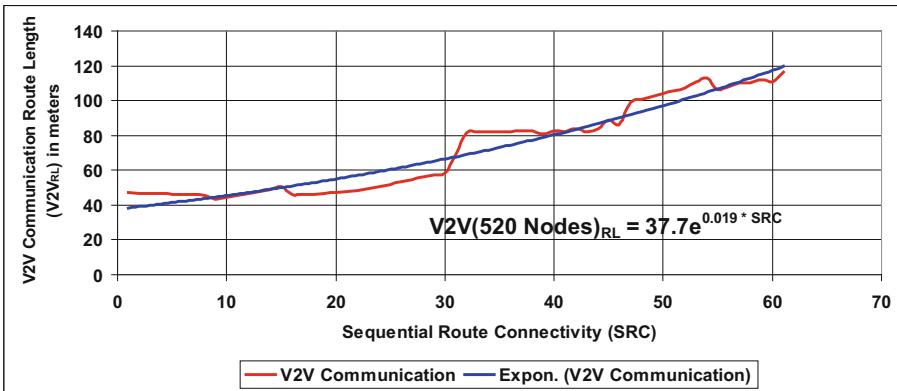


Fig. 4. Relationship between route length and connectivity over time for traffic density of 520 vehicles (nodes)

η : Route length parameter dependent on traffic density.

β : Scaling parameter.

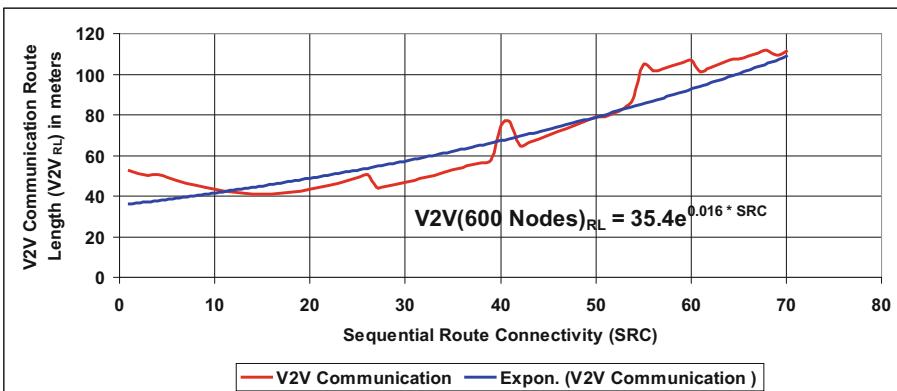


Fig. 5. Relationship between route length and connectivity over time for traffic density of 600 vehicles (nodes)

Figure 6, 7 and 8 present the relationship between network lifetime and sequential route connectivity (SRC) with exponential fitting curves. The relationship is decreasing exponential as it should be, since the number of hops and route length increases, causing a decrement in network lifetime as more energy is consumed through V2V communication. The fitted curve can be approximated by Eq. (8).

$$\text{Network Lifetime } (N) = \kappa * \exp(-\rho * \text{SRC}) \quad (8)$$

Where;

SRC: Sequential Route Connectivity.

N : Number of Vehicles (Nodes) in a spatial domain.

κ : Network lifetime parameter dependent on traffic density.

ρ : Scaling parameter.

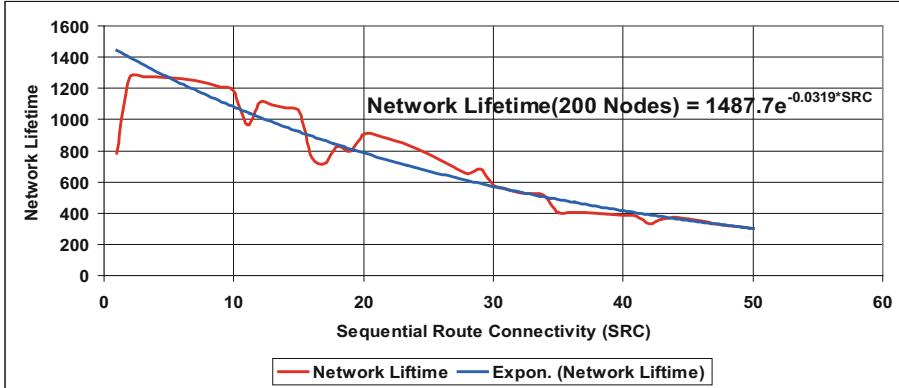


Fig. 6. Relationship between network lifetime and connectivity over time for traffic density of 200 vehicles (nodes)

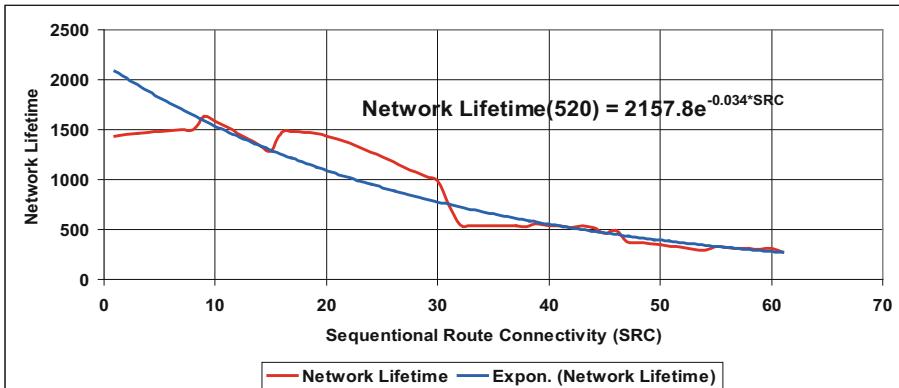


Fig. 7. Relationship between network lifetime and connectivity over time for traffic density of 520 vehicles (nodes)

By limiting the exponent in Eqs. (7) and (8) to one variable, and looking at ratios of values as a function of nodes, it is then sufficient to compare the values multiplied by the exponential function as shown in Table 4.

Network Life time supposed to decrease as number of nodes increases, based on the assumption that more BSMs are exchanged. However, if a congestion state occurs, then the number of BSMs will be reduced as the vehicle under consideration is boxed and surrounded by finite number of vehicles, hence, shorter routes of communication and fewer number of exchanged safety messages. Thus and from Table 4, we realize an increase in network lifetime and decrease in route length.

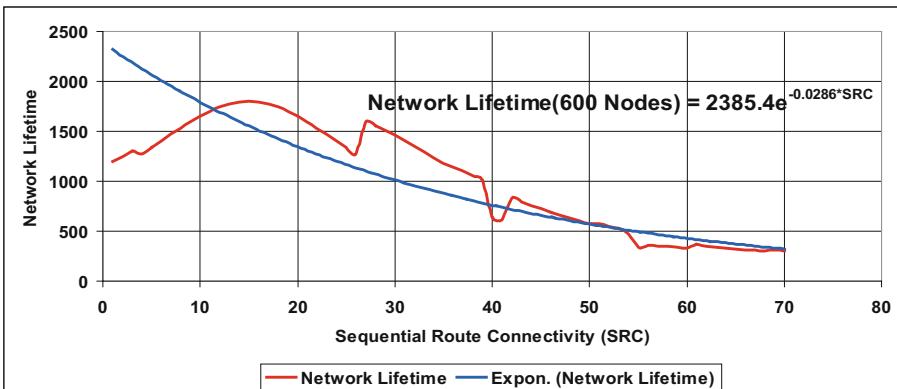


Fig. 8. Relationship between network lifetime and connectivity over time for traffic density of 600 vehicles (nodes)

Table 4. Testing nodes.

Number of nodes	V2VRL	Network lifetime
200	46.9	1487.7
520	37.7	2157.8
600	35.4	2385.4

To distinguish between heavy traffic and congestion (traffic standstill), the presented neural network model is used to learn from data in Tables 1, 2, 3 and 4 and to generate a prediction, classification, distinction, and confirmation of traffic status, depending on V2V communication and to provide three types of traffic states:

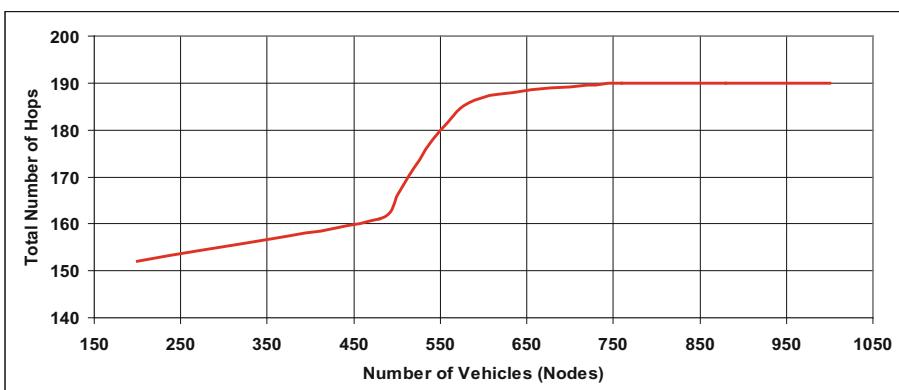
1. Smooth traffic flow
2. Smooth to Heavy traffic
3. Heavy to stand still traffic (congestion)

The obtained data is not descriptive of actual situation as it is predictive to avoid such conditions occurring by feeding such classification back to Traffic Management Center (TMC) and Traffic Operational Centers (TOC) in order to re-route traffic and avoid congestion by also controlling traffic lights. Such predictive data is shown in Table 5. The shown data in Table 5 proves the three mentioned cases, as when smooth traffic is flowing, there is a dynamic change in hops count, average route length, and average network lifetime, up to the point where regardless of the increase in vehicles, all other parameters stay constant at fixed values.

Table 5. Neural networks predicted data using WWEA.

Number of vehicles (nodes)	Hops count	Average route length	Average network lifetime	Traffic status
200	152	76	736	Smooth traffic
480	161	75	804	
500	166	74	846	Smooth to heavy traffic turning points
520	172	72	899	
550	180	69	973	
600	187	67	1027	Heavy to standstill traffic turning points
750	190	66	1044	
760	190	66	1044	
800	190	66	1044	Standstill traffic (congestion)
840	190	66	1044	
850	190	66	1044	
880	190	66	1044	
900	190	66	1044	
950	190	66	1044	
1000	190	66	1044	

Figure 9, 10 and 11 illustrate the predicted data in Table 5 with the three states of traffic (smooth traffic, smooth to heavy traffic, and heavy to standstill traffic) clear in the plots. The concavity of the turning point shows opposite differential signs to further support the predicted data.

**Fig. 9.** Predicted V2V route hops as a function of traffic density

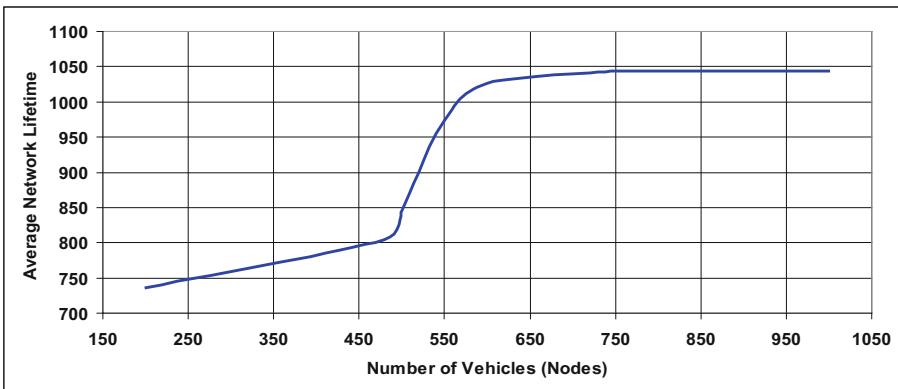


Fig. 10. Predicted V2V average life time as a function of traffic density

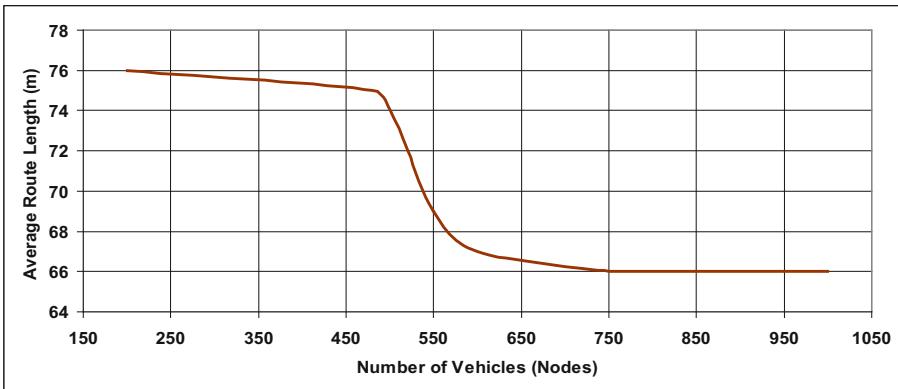


Fig. 11. Predicted V2V average route length as a function of traffic density

4 Conclusions

This work presents an intelligent approach to manage traffic through making use of V2V communication and the formed VANETs that connects vehicles. The neural network structure designed and based on Weigend Weight Elimination Algorithm (WWEA) is used to predict congestion using hops count, average network lifetime, and average route length as feeds to traffic controlling centers, whereby algorithms are used to better manage traffic and avoid congestion. Indirect correlation to network energy is shown through network lifetime. The results show a promising concept if further developed and applied will enable smarter management of traffic and vehicles. Explicit energy correlation will result in a more accurate prediction. Also, modelling of different scenarios such as accidents and work zones will be very helpful. Correlation of number of lanes and weather conditions in relation to threshold decision of congestion or no congestion will focus the predictive algorithm further, as vehicles speeds are weather dependent.

References

1. Baek, M., Jeong, D., Choi, D., Lee, S.: Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. *Sensors* **20**(288), 1–26 (2020)
2. Hussain, J.S., Wu, D., Xin, W., Memon, S., Bux, N., Saleem, A.: Reliability and connectivity analysis of vehicular ad hoc networks for a highway tunnel. *Int. J. Adv. Comput. Sci. Appl.* **10**(4), 181–186 (2019)
3. Li, T., Ngoduy, N., Hui, F., Zhao, X.: A car-following model to assess the impact of V2V messages on traffic dynamics. *Transp. B Transport Dyn.* **8**(1), 150–165 (2020)
4. Bai, Y., Zheng, K., Wang, Z., Wang, X.: MC-safe: multi-channel real-time V2V communication for enhancing driving safety. *ACM Trans. Cyber-Phys. Syst.* **4**(4), 1–27 (2020)
5. Guerber, C., Gomes, E., Fonseca, M., Munaretto, A., Silva, T.: Transmission opportunities: a new approach to improve quality in V2V networks. *Wirel. Commun. Mob. Comput.* **2019**, 1–20 (2019). Article ID: 1708437
6. Jung, C., Lee, D., Lee, S., Shim, D.: V2X-communication-aided autonomous driving: system design and experimental validation. *Sensors* **20**, 1–21 (2020)
7. Xie, D., Wen, Y., Zhao, X., Li, X., He, Z.: Cooperative driving strategies of connected vehicles for stabilizing traffic flow. *Transp. B Transport Dyn.* **8**(1), 166–181 (2020)
8. Chen, Y., Lu, C., Chu, W.: A cooperative driving strategy based on velocity prediction for connected vehicles with robust path-following control. *IEEE Internet Things J.* **7**(5), 3822–3832 (2020)
9. Mertens, J., Knies, C., Diermeyer, F., Escherle, S., Kraus, S.: The need for cooperative automated driving. *Electronics* **9**(754), 1–20 (2020)
10. Yu, K., Peng, L., Ding, X., Zhang, F., Chen, M.: Prediction of instantaneous driving safety in emergency scenarios based on connected vehicle basic safety messages. *J. Intell. Connected Veh.* **2**(2), 78–90 (2019)
11. Gao, K., Han, F., Dong, P., Xiong, N., Du, R.: Connected vehicle as a mobile sensor for real time queue length at signalized intersections. *Sensors* **19**, 1–22 (2019)
12. El Zorkany, M., Yasser, A., Galal, A.I.: Vehicle to vehicle “V2V” communication: scope, importance, challenges, research directions and future. *The Open Transp. J.* **14**, 86–98 (2020)
13. Baek, M., Jeong, D., Choi, D., Lee, S.: Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. *Sensors* **20**, 1–26 (2020)
14. Nampally, V., Sharma, R.: A novel protocol for safety messaging and secure communication for VANET system: DSRC. *Int. J. Eng. Res. Technol.* **9**(1), 391–397 (2020)
15. Kim, H., Kim, T.: Vehicle-to-vehicle (V2V) message content plausibility check for platoons through low-power beaconing. *Sensors* **19**, 1–20 (2019)
16. Sheikh, M., Liang, J., Wang, W.: Security and privacy in vehicular ad hoc network and vehicle cloud computing: a survey. *Wirel. Commun. Mob. Comput.* **2020**, 1–25 (2020). Article ID: 5129620
17. Liu, X., Jaekel, A.: Congestion control in V2V safety communication: problem, analysis, approaches. *Electronics* **8**, 1–243 (2019)
18. Seyd Ammar, A., Abolfazl, H., Hariharan, K., Farid, A., Ehsan, M.: V2V System Congestion Control Validation and Performance. *IEEE Trans. Veh. Technol.* **86**(3), 2102–2110 (2019)
19. Son, S., Park, K.: BEAT: beacon inter-reception time ensured adaptive transmission for vehicle-to-vehicle safety communication. *Sensors* **19**, 1–11 (2019)
20. Nahar, K., Sharma, S.: Congestion control in VANET at MAC layer: a review. *Int. J. Eng. Res. Technol.* **9**(3), 509–515 (2020)
21. Singh, H., Laxmi, V., Malik, A., Batra, I.: Current research on congestion control schemes in VANET: a practical interpretation. *Int. J. Recent Technol. Eng.* **8**(4), 4336–4341 (2019)

22. Zhang, T., Liu, S., Xiang, W., Xu, L., Qin, K., Yan, X.: A real-time channel prediction model based on neural networks for dedicated short-range communications. *Sensors* **19**, 1–21 (2019)
23. Elwekeil, M., Wang, T., Zhang, S.: Deep learning for joint adaptations of transmission rate and payload length in vehicular networks. *Sensors* **19**, 1–21 (2019)
24. Mignardi, S., Buratti, C., Bazzi, A., Verdone, R.: Path loss prediction based on machine learning: principle, method, and data expansion. *Appl. Sci.* **9**, 1–18 (2019)
25. Thrane, J., Zibar, D., Christiansen, H.: Model-aided deep learning method for path loss prediction in mobile communication systems at 2.6 GHz. *IEEE Access* **8**, 7925–7936 (2020)
26. Popoola, S., et al.: Determination of neural network parameters for path loss prediction in very high frequency wireless channel. *IEEE Access* **7**, 150462–150483 (2019)
27. Cavalcanti, B., Cavalcante, G., de Mendonça, L., Cantanhede, G., de Oliveira, M., D’Assunção, A.: A hybrid path loss prediction model based on artificial neural networks using empirical models for LTE and LTE-A at 800 MHz and 2600 MHz. *J. Microwaves Optoelectron. Electromagn. Appl.* **16**(3), 708–722 (2017)
28. Popoola, S., Adetiba, E., Atayero, A., Faruk, N., Calafate, C.: Optimal model for path loss predictions using feed-forward neural networks. *Cogent Eng.* **5**, 1–19 (2018). Article: 1444345
29. Weigend, A., Rumelhart, A., Huberman, B.: Generalization by weight-elimination with application to forecasting. In: Advances in Neural Information Processing Systems Conference, pp. 875–882. ACM (1990)



Urban Planning to Prevent Pandemics: Urban Design Implications of BiocyberSecurity (BCS)

Lucas Potter¹, Ernestine Powell^{2(✉)}, Orlando Ayala³, and Xavier-Lewis Palmer^{1,4}

¹ Frank Batten College of Engineering and Technology, Biomedical Engineering Institute, Old Dominion University, 4111 Monarch Way, Norfolk, VA 23508, USA
Lpott005@odu.edu

² Department of Neuroscience, College of Natural and Behavioral Sciences, Christopher Newport University, Forbes Hall 1053, Newport News, VA 23606, USA
ernestinepowell@protonmail.com

³ Department of Engineering Technology, Frank Batten College of Engineering and Technology, Old Dominion University, 102 Kaufman Hall, Norfolk, VA 23529, USA

⁴ School of Cybersecurity, Old Dominion University, Norfolk, VA 23529, USA

Abstract. Recent pandemic complications have spurred significant conversation concerning how various organizations rethink their infrastructure and how people interact with it. More specifically, some city governments are considering how they can improve or re-purpose infrastructural design as they manage pandemics and potentially those to follow. In facilitating this effort, it is essential to look at effectively established principles and perspectives of modern designers to forge new functional and practical infrastructure that can accommodate behavioral changes within a pandemic. As smart cities increase in number, this is of particular concern due to their utilization of computing components in routine citizen and infrastructural interactions. This prevalent interface of citizens interacting with cyber-physical and remote cyber systems requires consideration of the intersection of biosecurity, particularly during a pandemic. Consequently, in this work, a proposed baseline by which city planners can draw inspiration for their efforts to boost resilience to pandemics is designed using First Principles, along with a collection of modern opinions. This exploratory work in progress aims to spur such questions in literature and provoke meaningful dialogue towards this endeavor.

Keywords: Infrastructure · Biocybersecurity · Cyberbiosecurity · Urban Bio-Security

1 Introduction

Cities have long been shaped by the needs of community health from ancient times, even to today. For instance, Roman baths were a focus, and health was a consistent feature of the seminal “Ten Books on Architecture” [1]. In another example, epidemics that struck the urban environments between the 18th and 20th centuries substantially formed the modern urban setting. A more specific example from this period stems from

the beginning of the 20th century in Senegal [2] with attempts to ameliorate conditions like heart disease with urban planning [3]. In stark contrast, today, the collaboration and progress of public health and urban planning have become notably disparate, particularly regarding the spread of diseases. Ironically, two fields that began with noble intentions of decreasing preventable diseases face a threat found in the current pandemic that exhibits both fields' flaws [4]. This occurrence is especially odd, considering that some consider epidemics to be the engine of urban planning [5]. However, it is time to acknowledge two unique facts of the world as it is today that exacerbate the spread of biological agents during a pandemic.

Foremost is the massively sizable urban population in modern times that exponentially increases the number of possible interactions with people and their intersections compared to previous eras. The urban population was “54% of the total global population ... and is expected to grow approximately 1.84% per year between 2015 and 2020, 1.63% per year between 2020 and 2025, and 1.44% per year between 2025 and 2030” [6]. The second contributing fact is the sheer breadth of interconnectedness that exists in today’s world. The prescient Arthur C. Clarke mentioned future potential changes in urban design in 1964, with uncanny accuracy [7]. Clarke proposed that a virtual means of personal interaction would arise from new wireless technologies, which are now known as the Internet. He also discussed how this technology could prove advantageous as the world adapts to a new normal for its interactions to confront the advent of viral pandemics [7]. Concerning the advent of COVID-19, communities around the world faced complexities embedded within balancing the need to efficiently preserve both community health and the economy by oscillating between implementing lock-downs and lifting said restrictions. Be the lift gradual or immediate, many communities face difficulties that improved urban planning can potentially mitigate by addressing the new reality of living in an urban environment with a hardy, airborne disease.

This paper explores a possible urban planning effort and design notions that can be considered for more universal implementation for the current pandemic and for years beyond COVID-19. Additionally, digital technologies are not discussed at length within this paper due to this work being a supportive piece aimed towards those with background knowledge of such technologies. It is prudent that data, biometric and peripheral, collectible and accessible throughout, is kept in consideration for discussion.

2 Background

The most important thing to remember as we embark on our quest to design a disease-resistant city is that the vectors of diseases exist in both the temporal and spatial domains. For us, that means that an effective urban design would confirm the ability to separate potential attack vectors according to space by increasing the physical distance and increasing separation by time. This involves not only increasing gaps between public transit seating or having fewer spaces in restaurants, but also increasing the time between different individuals entering that space. In practice, this could mean having such a large number of public transportation options that only a certain number of people would use the same seat before it was adequately sanitized. To be clear, creating mechanisms to control the spread of disease in cities is hardly a novel concept. The formation of specific zoning boards, health inspections, and metropolitan boards have all been in response to

the spread of diseases in the urban environment. As exemplified in the recent past, cities have often taken the initiative to control spread, although sometimes in a manner that further harmed relationships between the government and minority groups [8]. These swift changes, such as the rat-catching and sanitation initiatives encouraged by the local government during the San Francisco Bubonic Plague epidemic and using sewage to help track the number of Covid cases, demonstrate the innovation that arises when cities are forced to handle these crises [8, 9]. Furthermore, the essence of cities as gathering places could not be changed from their ancient roots as much as they can be changed in the modern world. That is to say that the health considerations of a city have nearly always been seen as side issues to the primary objective of keeping the city's mechanisms running and its economy strong. Again, this can be seen in the San Francisco government's strong denial of the Bubonic Plague, so as to better maintain trade relationships with other states and countries [8, 10]. Eventually, the plague spread outside of Chinatown and affected a wider demographic than before with increasing case numbers, forcing the local government to no longer suppress efforts to acknowledge the existence of the epidemic and cease suppressing efforts to resolve it [8]. This continuous prioritizing of a city's economy in urban design or function over the well being of a city's citizens may be an error in judgment to be reconsidered.

3 Methods

For this paper, first, the authors gathered opinions and sought inspiration from past differences in urban design. This ranged as far back as the Roman Empire [1], to the Cholera epidemics of the 1800s [1], and even to the last major cases of bubonic plague [10]. Second, we had to acknowledge that the future disease vectors that the urban setting may face might potentially stem from not only naturally occurring diseases, but also diseases designed by individual hostile actors or malicious groups, or even nation-states. In this case, we felt the need to acknowledge the possible vectors which, while not seen as a natural or conventional way for a disease to enter the population, could be co-opted to deliver the microbe.

4 Results

In the case of creating a disease-resistant urban environment, we separated the task into several components. This honors the complexity of this task by reflecting the conceptualization of urban environments as not just spaces, but the combination of processes that form them.

The first component was to identify the present potential attack vectors that could be used to infect a given urban environment. Yet, this is not to say that these vectors have or will be used to introduce disease into the urban environment. In most cases, the vectors in question are simply routes of entry to bodies that are not native to the urban environment. Consider the simple example of food. Few cities could cultivate their crops on a scale necessary to feed their inhabitants. Yet, carefully designed and regulated vertical farms may resolve this, but such efforts remain to be seen. Regardless, upon such a route being viable, a city can consider how the food produced and later consumed could be a route

for harmful biological agents if the facilities in question are not vigilantly and properly monitored. Notable and excellent examples include farm culling and factory recalls of meats.

The second component questions the relevant principles upon which cities are based. This includes our current understanding of what it means to be in an urban environment. In essence, it is a framework of developing reasons as to why people choose to live in an urban environment, and how those needs could be met in a more thoughtfully designed space. These choices can vary along lines of age, work, and more. For example, some employees may choose urban environments for higher pay, while some employers may do so for access to a diverse and robust supply of workers. Other individuals who choose along non-economic aligned angles may do so for aesthetic reasons or reasons aligned with other cultural groupings. Real-world examples include college or industrial parks that capture a wide range of citizen preferences [9].

The next component involves creating a system of responding to a pandemic-level biological threat in an urban environment. This is the culmination of the first two steps: the rigorous threat analysis of an idealized city, and the questioning of why the urban environment exists the way it does now. To further elaborate, city weaknesses and strengths, such as the age, grade, and integrity of the infrastructure, as well as the reasons behind its composition, are important to consider. These considerations can in turn empower planning how to leverage the strengths and mitigate the weaknesses of urban environments, especially when implemented into official policy with active effort and effective documentation [11].

Taking these components into consideration, in Table 1, we suggest a set of rules through which a disease-resistant city could be created. We additionally included guidance as to what such a city may look like. Below a list of potential attack vectors is given.

4.1 Potential Attack Vectors

To begin this stage of our analysis, the authors hypothetically assumed the role of the biological antagonists, to simulate exploitations of vulnerabilities in infrastructure. The goal of this exercise would, at the least, imagine how a malicious agent or a naturally occurring, but contaminating biological agent might eventually immobilize a city and cause outstanding biological damage to urban structures. This in turn could thereby disable significant parts of a city, negatively impacting the population of an urban environment. This in turn presents discussion material for city planners to address in order to protect the public.

It helps to differentiate a biological agent from the conventional ordinance, for understanding its effects on an environment. Vital importance in the capacity of a biological agent lies within its unlimited resource as a weapon, particularly as a canon in need of something to shoot. A piece of ordinance, once expended, typically no longer presents a threat. Yet, a virus or microbe can continuously infect a population long after the original delivery has been completed so long as it has access to hosts or a viable means of replication. This does rely on the biological agent not being fatal to the host before successful transmission and infection of another person. Keeping this in mind, the goal no longer becomes simple delivery, but the targeting and dissemination of the microbial

Table 1. Potential attack vectors

Common item	Purpose(s)	Means of vulnerability/exploit
Bench	Place to sit	Particles on surfaces
Bus	Transportation	Aerosols ejected from cushions as they are sat upon, provided that they can settle and become embedded without loss of function Some buses have electronic kiosks through which pay is transferred, which can be a route of transfer if they are interacted with through skin contact (Thus, Pathogen transfer through skin from improper sanitation)
Toilets	Restroom relief	Aerosols via a toilet plume
Active-Equipment (especially outdoor)	Fitness	Pathogen transfer through skin from improperly sanitation
Accessibility Accommodations (Railings, Motorized Scooter Rentals, Wheelchairs, Patient Call Bells, Kiosks)	Assist with individuals having more autonomy to complete their activities of daily living and assist with their quality of life	Pathogen transfer through skin from improper sanitation
Phone Booths	Communication	Lingering Aerosols in the enclosed space, resulting in a "hot box"-like phenomena
Pool	Relief from hot temperatures, low-impact exercise	Pathogen transfer through skin or orally, through ingesting of pool water and mixed fluids
Hand Dryers	Replacement for towels	Particles aerosolized by high-velocity air (In the case of motion detected air dryers, this is harder to control as improper detection can spur aerosolization at less controllable rates)
Public, Open Computers and Kiosks (ATMS, Information Booths, and Similar)	Accessing Information, money, or services	Pathogen transfer through skin from unclean surfaces or harvesting of biometrics (fingerprints, voice)

payload. In this case, urban planners must strive to understand the complexity of the problem. A person in a major metropolitan city could potentially be within the infection

range of thousands of people during their day. This is all without taking into account any changes in virulence, the severity of the disease, the contagiousness, the ability of the healthcare system and government to manage the infected individuals, the ability and willingness of facilities and individuals in helping to reduce the spread and practice proper disease control measures, or other means connected to the spread of diseases in an urban environment.

This also has major implications in the delivery of medical services in an urban environment. For instance, say that a disease is virulent enough to cause a city's major hospitals to send all other patients home to self-quarantine, leaving the hospitals free to deal with the patients infected with a single disease. The only other avenue of delivering medicine would be local pharmacies, pop-up treatment centers, and similar authorized distribution facilities. In this case, those local pharmacies, while they would be more diffused, present a much more vulnerable target for potential hostile actors as they contain less expensive equipment and patient records, per building to protect. The resources needed to purposefully infect and endanger those pharmacies would be larger, as the target is distributed and requires more resources. However, if this avenue was already composed of asymptomatic individuals or people with mild symptoms, then the problem of dissemination solves itself.

Likewise, public transport, such as trains, trams, or buses, offers another route of spread for the biological agent. This can be unintentional due to close contact or improper sanitation in an appropriate time frame. It could also be intentional if the agent is stable enough to be planted in these public spaces. Additionally, biological agents can be intentionally spread via deliberate infection of gas stations or similar nodes at which transporters restock or relieve themselves. The primary issue, in this case, is that to support the current quality of life in an urban environment, a massive quantity of goods are needed, which are not currently cost-effective to produce locally in the present scheme of resource allocation. Each means of resource transport is more critical than the last and the ability to halt a city's operations by severely affecting even a fraction of these resources can reduce or decimate the quality of life in urban environments. Thus, vigorous protection is of the utmost importance.

To best account for insidious methods of spread, we considered likely ways to accomplish this task. Infection of surfaces that one contacts is a common concern, but is hardly the most insidious method. It could be partially addressed by the implementation of simple cleaning measures. Still, this remedy is vulnerable to a natural aspect of city life – wealth inequality. As many communities saw early during the COVID-19 pandemic, much publicly accessible infrastructure was ill-equipped to deal with a pandemic. For example, efforts focused on cleaning of public transportation and seating, such as subways and park benches, were unintentionally countered through frequent contamination of surfaces via direct touching of surfaces or expelling of particles via coughing or talking by infected individuals [12–15]. The volume and frequency of such are difficult to stem given that so much of this unintentional contamination can be attributed to people traveling for work who do so out of necessity, unintentionally and partially counteracting lock-down as workers without access to adequate social safety nets simply cannot afford to stop working [16]. Even during a crisis, few of them received sufficient financial support that prevented them from becoming unintentional deliverers of a viral payload. This

was and is amplified in cases where workers lack(ed) proper personal protective equipment [17]. Notice of this is particularly important within places where this is perhaps needed the most: hospitals, nursing homes, other long-term or specialized care facilities, and practically anywhere with front-line essential personnel [16]. An infection could be easily encouraged by deliberately infecting the poorest workers of a city, especially without their knowledge. In particular, the individuals who are directly responsible for sanitation procedures can be at an even higher level of risk.

Even sanitation could be co-opted into a method of negative efforts for preventing the spread, as resources are wasted where they could be better spent on other infrastructural improvements or treatment. This is also to further ask, without addressing these base issues, what frequency of sanitation is even useful? To what degree is crowd control helpful in making even that number useful? For example, if a homeless person was to sleep in a train right after it gets sanitized what is the point of sanitizing it in the first place? What could be done to motivate governments to properly house the homeless to prevent this? A mere redesign of surfaces to make touching them unattractive, found most often under the term “Hostile Architecture”, has often shown to be counter-intuitive due to human ingenuity in times of desperate situations [18–21]. Reducing surfaces to touch, as well as funding smart, touch-less interfaces, warning devices, or monitoring systems that address the most critical of spaces to sanitize (areas that are the site of inordinate exposure to body fluids) may help address concerns in these spaces [15].

Nonetheless, as it currently stands, waste service professionals have little choice to continue doing their jobs amid a pandemic. This is especially true in cases without proper representation advocating for workers’ health. We term this idea – of the infection of the poorest members of a population, especially in a city where wealth inequality is large, a trickle-up attack. The primary concept is that while wealthy members of a population may have access to the resources to combat a biological agent with longevity, the poorer citizens may not. However, since one cannot live within a bubble in a city, eventually the poorer citizens might infect the wealthier members, leading to deeper penetration of the agent in the population.

There is a commonality to the problem considered above. Why, exactly, do large populations want to put themselves at such risk by living so tightly together? The answer is likely too complex to be answered in an exploratory and commentary paper such as this, but a simple assumption is that humans are typically socially-centered and desire interaction with one another [22]. This interaction can take the form of either structured, purposeful groupings (companies, political organizations, or public services) or informal groupings (sporting or music events). To better understand how to view those interactions as a potential vector, we can turn to design from first principles [17].

4.2 Designing from First Principles

In this context, Principles of a City are:

- 1) to centralize resources in order to lay claim to a geographic area
- 2) to provide services to residents in an urban environment
- 3) to develop technologies, ability, or knowledge.

In this exercise, one first explores the First Principle of a city, which is to control a given geographic area by political means. In the case of a globally connected world, the presence of disease can quickly be spread from around the globe. The transition from traveling strictly on foot or via the assistance of an animal, to utilizing trains, cars, and planes has rendered the impact of geographic distance and features to be less of a concern to the spread of disease and more of a catalyst. Just one infected passenger is all that is needed to kickstart the spread of infection to another nation, which is why infrastructure capable of detecting, containing, and stopping the spread of infections remains important as the essential idea of distance or geographic features being able to inhibit or excite the spread of disease thus becomes negligible [8, 12, 13, 23]. Which poses the question: should the borders of regions, states, or even nations be rethought? For instance, the divisions of most states in the United States are either built upon relatively arbitrary geographic divisions of latitude and longitude, or divisions based on geological features. In post-colonial states like much of central Asia and Africa, some of these divisions exist in an even more arbitrary way, such as the reasoning behind the partitioning of the Ottoman Empire [18]. Why not make the divisions in such a way as to enable the ease of quarantine? This is not meant as a way to return to more insular communities, but rather as a way to corral the spread of diseases by exciting more local trade, with global trade still existing, but not being the lifeblood of any one community. Take this for an example- imagine two fictional states: East and West Dakota. Both of these states are primarily self-sufficient, but East Dakota lacks ready food supplies, and West Dakota has no energy capability. In the case of an emergent threat response, but especially in the case of a pandemic, why should these two states not cooperate? To a medical officer, does this political division do anything to aid them? Then why ask a medical professional to, if not tolerate it, have to work around it, instead of letting them provide medical support as best they can? Yes, hospital policies, as well as local, state and federal laws related to licensure impose restrictions concerning where medical professionals can work with legal and career consequences should they violate any laws. However, perhaps in times of crisis, such as during pandemics, there could be regulations made that better allow for monitored orchestration of aid across these borders and barriers. This has been the case across the U.S. by way of varying waivers, regulated by the states, to the telehealth telephone and video conference license requirement and waivers that allow physicians from different states to collaborate with each other [17]. This is especially helpful for locations with more Covid experience sharing knowledge with locations that have less or little experience [24, 25]. This network can also be a saving grace for healthcare facilities that are in dire need of supplies, if other facilities are able and willing to donate materials, especially personal protective equipment or PPE [26].

Now, we turn to our second principle. Cities are meant to serve the people in them, to enable the specialization of citizens. Expecting a neurosurgeon to start his day by harvesting his or her wheat would be a difficult demand to ask. However, some of these services are now found even in rural areas. Telecommunications enables the spread of news and novelties. Sanitation is found at similar levels around the developed world. Transportation can link rural and urban areas alike. This leads to the delivery of items being comparable in both environments. This is to say that many of the most essential

services are available now even without participating in the urban environment. Even business, typically done in crowded urban marketplaces, has made a transition in part to the internet enabled world. In this case, assuming the appropriate resource allocation to services like sanitation, communications, and consumer goods, the second principle of cities is negated in part in developed countries.

The matter of using cities as hubs for commercial activity brings up another interesting point that concerns our third principle. Some believe that we are currently in an “Idea Economy or Economy of Ideas”, where concepts or thoughts are the most valuable commodities to the world’s economic system [27, 28]. Supporting this is the current trend with placing emphasis on innovative yet effective solutions due to the lack of or decreased productivity in several fields, particularly research, transportation, and architecture [19, 29]. This has in turn spurred an increase in funding or re-analysis of how to improve methods for the benefit of the city and its citizens [19, 29]. In this case—why are we bothering to spread ideas in such an archaic manner if virtual means suffice and have experienced success? Telemedicine has shown this possible to a certain extent in the medical field [16]. The rise and growth of companies like WebEx and Zoom have become testament that much work, even in research, can continue without the need for traditional, in-person means of idea exchange [27, 30]. For instance, many companies discovered during the different quarantine orders that just as much work, if not more can be completed even without access to a physical space, or the physical presence of people in the same space. Also, many government research entities work closely with facilities across the county; and the adoption of ARPANET showed the potential of these research facilities to work within a virtual framework [31]. Also noteworthy, many workers have been found to be more productive working remotely [32]. Thus, lending support to the idea that a virtual partnership or exchange is well-founded. The adoption of a completely virtual research group is a fascinating idea that may be the future of innovation. As an example, the very work you are reading is a work that required no physical presence on the part of any of the authors.

All of these principles, however, neglect the comprehensive reality of life in a city. While many services are done by public servants that are ultimately accountable to citizens, many features of a city are through privatization. This leads to another inquiry worth pondering. Should a private organization, whose ultimate purpose is profit, be allowed to make decisions that negatively affect employee’s health and well-being? More importantly, should it be allowed to make decisions that affect the health and well-being of individuals that are not compensated by that company, such as community members who are in contact with their workers and who may or may not be contagious? This is a question for each government to answer as each individual will no doubt make contact with the infrastructure within the urban environment, creating a plethora of opportunities for the spread of infection.

Now, though the formerly addressed principles are important, some sub-principles also deserve mention. Another important aspect of a city is political thought, specifically through political demonstrations. One of the undeniable powers of a centralized location of a population is the ability to instigate great societal change in a relatively short amount of time, either through dialogue, protests, riots, or demonstrations. However, in the case of a pandemic situation, a large group of people can be seen as not just an example of

political will, but as a distributed denial of service (DDOS) attack in a physical space. This scenario assumes two things- that the group willingly gathers together, and that their motivation for doing so overpowers their motivation for safety. The former case (willingness to gather) could be a condition of income or labor relations. For instance, a group of factory workers may not be fiscally able to stop working. Does this mean that to prevent the spread of pandemics, paid sick leave should be a mandated guarantee? For that matter, should an amount of personal space be guaranteed, or required, to build or rent a home? The latter case assumes a pained motivation behind such a gathering. What if the group is forced together by circumstance? Then, should the amount of space a citizen of a city can expect be regulated in the defense of public health? If a city does mandate a minimum amount of space that a citizen is to be guaranteed, what happens to those citizens that fall in the cracks – the vagrants and those without homes? It is important to consider how design can motivate proper spacing so that such individuals can not only group safely, but also so that they are not encouraged to leave behind fomites.

4.3 Hypothetical Responses to Disease

One of the ideas that came to mind when exploring the aforementioned principles of urban design were the ways that our cities respond to disease - and so the people that treat it. In this case, what should be planned when the first responders of a city become the vector of a disease? This is especially the case during the middle of the Covid-19 epidemic, when PPE shortages prevented medical providers from adequate protection and high infection rates among health workers placed doubt on the reasonable capacity for mass quarantining without significant losses in workforce output [23, 33, 34]. To what degree could telepresence alleviate some of this stress? Further, to what degree may disruptions occur, through natural and malicious actions, that need to be additionally safeguarded? Of course, future designs of medical robotics could, theoretically, enable the complete care of a patient from transport to treatment, but that is not a discussion for most citizens as of now and is not a financial consideration for more underserved hospitals, or even healthcare systems. As seen during this last epidemic, many elective surgeries have been canceled or postponed, and telemedicine has surged in use for more routine visits. The latter is likely to see a short- term sustained increase as the public moves to limit their liability and cut costs. However, the goal of most telepresence (to offer consultation support and treatment or guide their counterparts through procedures) is hampered by one massive consideration – that of privacy. As discussed in a previous work [35], telepresence devices could be exploited in ungainly ways to make them incompatible with the ideals of patient privacy, requiring re-designing telepresence practices in some instances, in the future; it is important to be mindful about how current practices may change. This is not currently an issue, for most commonly accessible, personal gadgets that allow a user to track their health are devices like pedometers and sleep trackers, which are of minimal importance. However, as more and more data becomes available, the commensurate risks grow. Perhaps an easier method could be to have citizens of a city become allies in preventing the disease. Imagine creating a smartphone app that plays a high-pitched, aggravating tone when in less than 2 m of another smartphone.

Another imagination gives way to infrastructure that alarms based on proximity to certain devices that present a risk to the user or others in proximity to it. Larger entities such as companies or governments could also incentivize larger regional efforts to curb the spread of covid-19 as based on tracking through waste or trash deposited [36]. Efforts such as these could prevent, at the least, the most immediate risk factors of infection. These are, but a start, but worth considering and building upon.

5 Further Discussion

The relationship between humanity and its technology is one of the most interdependent relationships that exists. Our heroes and villains, kings and commoners, poets and scientists – all are defined by what they have, what they create, what they design, and what they use, leading to the change of themselves and their environment. Such applies to all individuals in their varying uses of technology, especially with planned and dynamically generated means of infrastructure. COVID-19 has shown the world that there is much change needed to phase out the risks that pandemics pose towards the use of urban infrastructure. Certainly, the current technology has prevented many deaths that would have occurred compared to prior pandemics centuries ago – the quality and amount of medical knowledge, training, and tools in such a short amount of time during our response to this crisis has likely never before been leveraged on this planet. And yet, it seems that the solutions to making our cities more resilient and more sustainable in defiance to this disease are not solely technological innovations given the losses incurred despite the use of our current technologies. Instead, it would seem that a new, more equitable social contract between people in our world may be the next best way, vaccine aside, to prevent such a scourge from occurring once more. It is questionable that there will ever be a single drug or injection that alone will obviate the need for medical treatments, or make pandemics a thing of the past. Instead, it may be that the nature of the disease in the urban environment relies not on our mastery of the concrete jungle, but on our social response to those who navigate it and answer why they use it at disadvantageous times. Our optimal response points to a holistic design and management of current and planned designed spaces that are mindful of our fellow citizens and their capacity to adapt, insofar as we can assist.

6 Limitations

Clear limitations exist in that this paper is not at all comprehensive of the intersection of BCS and urban design. Further, there exists expansive commentary and development upon first principles worth consulting that is not included herein. Moreover, detailed mechanisms of technologies discussed and how one may optimally use them, and beneficial or malicious purposes are not expounded upon within either as such is beyond the scope of this work and consequently requires more careful writing. Lastly, proposed designs may be effective on paper, but when adding in human free will and government politics, bringing the proposed plan to reality and fully realizing the potential positive benefits may not occur in the way envisioned. This paper is both commentary and exploration, but not more than such.

7 Conclusions

What one can make of this is that Covid-19's arrival and contemplation of other potential pandemics challenges us to rethink our urban planning and economic functions to adequately deal with pandemics. This requires us to map out potential attack vectors and be creative when doing so in order to reduce what natural and direct means may likely evade both counter and capture. Further, we must reconsider how we design and modify our cities based on first principles, considering the basic aspects of what makes a city and working from there. Ultimately, we must then redesign our responses to disease outbreaks. The disruption produced is the new normal and is likely to remain that way unless we can adapt from reflection on first principles to design a new normal to which we more willingly consent and frame as progress towards a more well prepared and managed city. It is hoped that this will provide a helpful basis for consideration in addressing urban planning designs.

8 Future Work

Future work entails a more robust discussion of vectors and mapped opinions of those in the area of Biocybersecurity/Cyberbiosecurity. Specifically, this would be the subject of a comprehensive review of recent papers in BCS at the intersection of Urban Design. As the evolution of urban design has included new technologies and threats that have emerged, those at the intersection of BCS will undoubtedly be no exception.

References

1. Pollio, V., Morgan, M.H., Warren, H.L., Pollio, V.: *The Ten Books on Architecture*. Harvard University Press, Columbia (2018)
2. Bigon, L.: A History of Urban Planning and Infectious Diseases: Colonial Senegal in the Early Twentieth Century, 21 February 2012. <https://www.hindawi.com/journals/usr/2012/589758/>. Accessed 13 July 2020
3. Barton, H., Tsourou, C.: *Healthy Urban Planning*. SPON Press, London (2000)
4. Corburn, J.: Confronting the challenges in reconnecting urban planning and public health. *Am. J. Public Health* **94**(4), 541–546 (2004). <https://doi.org/10.2105/ajph.94.4.541>
5. Malone, C., Bourassa, K.: Americans Didn't Wait for Their Governors To Tell Them To Stay Home Because of COVID-19, 8 May 2020. <https://fivethirtyeight.com/features/americans-didnt-wait-for-their-governors-to-tellthem-to-stay-home-because-of-covid-19/>
6. Knorr, D., Khoo, C.S.H., Augustin, M.A.: Food for an urban planet: challenges and research opportunities. *Front. Nutr.* **4**, 73 (2018)
7. Middleton, K.: Arthur C Clarke predicts the internet in 1964 [video]. YouTube, 22 December 2013. <https://www.youtube.com/watch?v=wC3E2qTCIY8>
8. Tirachini, A., Cats, O.: COVID-19 and public transportation: current assessment, prospects, and research needs. *J. Public Transp.* **22**(1) (2020). <https://doi.org/10.5038/2375-0901.22.1.1>. <https://scholarcommons.usf.edu/jpt/vol22/iss1/1>
9. Marohn, K.: Tracking COVID-19 through sewage, 22 May 2020. <https://www.mprnews.org/story/2020/05/22/meet-the-minnesota-scientisttrying-to-track-covid19-spread-through-sewage>. Accessed 06 July 2020

10. Tansey, T.: Plague in San Francisco: rats, racism and reform, 24 April 2019. <https://www.nature.com/articles/d41586-019-01239-x>. Accessed 20 Feb 2021
11. Laurian, L.: Planning for active living: should we support a new moral environmentalism? *Plann. Theory Pract.* **7**(2), 117–136 (2006)
12. Ayenew, B., et al.: Challenges and opportunities to tackle COVID-19 spread in Ethiopia. *J. PeerScientist* **2**(2), e1000014 (2020). <https://www.peerscientist.com/volume2/issue2/e100014/Challenges-and-opportunities-to-tackle-COVID-19-spread-in-Ethiopia.pdf>
13. Bonful, H.A., Addo-Lartey, A., Aheto, J.M.K., Ganle, J.K., Sarfo, B., Aryeetey, R.: Limiting spread of COVID-19 in Ghana: compliance audit of selected transportation stations in the Greater Accra region of Ghana. *PLoS ONE* **15**(9), e0238971 (2020). <https://doi.org/10.1371/journal.pone.0238971>
14. Swanson, M.W.: The sanitation syndrome: bubonic plague and urban native policy in the Cape Colony, 1900–1909. *J. Afr. Hist.* **17**(3), 387–410 (1977)
15. Ventura, J.R.: Virginia Residents Required To Wear Face Masks In Public Indoor Places, 27 May 2020. <https://www.ibtimes.com/virginia-residents-required-wear-face-masks-public-indoor-places-2983347>
16. Larochelle, M.R.: “Is it safe for me to go to work?” Risk stratification for workers during the Covid-19 pandemic. *New Engl. J. Med.* (2020). <https://doi.org/10.1056/nejmp2013413>
17. Federal Association of State Medical Boards: U.S. States and Territories Modifying Requirements for Telehealth in Response to Covid-19, 5 February 2021. <https://www.fsmb.org/sites/advocacy/pdf/states-waiving-licensure-requirements-for-telehealth-in-response-to-covid-19.pdf>. Accessed 10 Feb 2020
18. Baer, R.: See no evil: the true story of a ground soldier in the CIA’s war on terrorism (Reprint ed.). Broadway Books (2003)
19. Bloom, N., Jones, C., Van Reenen, J., Webb, M.: Great Ideas Are Getting Harder to Find: Unless we keep raising research inputs, economic growth will continue to slow in advanced nations, 20 December 2017. <https://sloanreview.mit.edu/article/great-ideas-are-getting-harder-to-find/>. Accessed 10 Feb 2021
20. Chellew, C.: Design paranoia. *Ontario Plann. J.* **31**, 18 (2016)
21. Petty, J.: The London spikes controversy: homelessness, urban securitisation and the question of ‘hostile architecture.’ *Int. J. Crime Justice Soc. Democr.* **5**(1), 67 (2016)
22. Chen, Y., et al.: High SARS-CoV-2 antibody prevalence among healthcare workers exposed to COVID-19 patients. *J. Infect.* **81**(3), 420–426 (2020)
23. Shen, J., et al.: Prevention and control of COVID-19 in public transportation: experience from China. *Environ. Pollut.* 115291 (2020). <https://doi.org/10.1016/j.envpol.2020.115291>
24. Krishnamoorthy, Y., Nagarajan, R., Saya, G.K., Menon, V.: Prevalence of psychological morbidities among general population, healthcare workers and COVID-19 patients amidst the COVID-19 pandemic: A systematic review and meta-analysis. *Psychiatry Res.* **293**, 113382 (2020)
25. Kumar, R., Nedungalaparambil, N.M., Mohanan, N.: Emergency and primary care collaboration during COVID-19 pandemic: a quick systematic review of reviews. *J. Fam. Med. Primary Care* **9**(8), 3856–3862 (2020). https://doi.org/10.4103/jfmpc.jfmpc_755_20
26. Sen-Crowe, B., Sutherland, M., McKenney, M., Elkbuli, A.: A closer look into global hospital beds capacity and resource shortages during the COVID-19 pandemic. *J. Surg. Res.* **260**, 56–63 (2021)
27. Barlow, J.: The economy of ideas, 1 March 1994. <https://www.wired.com/1994/03/economy-ideas/>
28. Paneth, N., Vinten-Johansen, P., Brody, H., Rip, M.: A rivalry of foulness: official and unofficial investigations of the London cholera epidemic of 1854. *Am. J. Public Health* **88**(10) (1998). <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1508470/pdf/amjph00022-0105.pdf>

29. Bloom, N., Jones, C., Van Reenen, J., Webb, M.: Are ideas getting harder to find? *Am. Econ. Rev.* **110**(4), 1104–1144 (2020). <https://doi.org/10.1257/aer.20180338>
30. Omidi, M.: Anti-homeless spikes are just the latest in ‘defensive urban architecture’. *The Guardian*, 12 June 2014. <https://www.theguardian.com/cities/2014/jun/12/antihomless-spikes-latest-defensive-urban-architecture>
31. Rudan, I.: A cascade of causes that led to the COVID-19 tragedy in Italy and in other European Union countries. *J. Glob. Health* **10**(1) (2020). <https://doi.org/10.7189/jogh.10.010335>
32. Hunter, P.: Remote working in research: an increasing usage of flexible work arrangements can improve productivity and creativity. **20**(1) (2019) <https://www.embopress.org/doi/full/10.15252/embr.201847435>
33. Chen, B., et al.: Basic psychological need satisfaction, need frustration, and need strength across four cultures. *Motiv. Emot.* **39**(2), 216–236 (2014). <https://doi.org/10.1007/s11031-014-9450-1>
34. Groat, L.N., Després, C.: The significance of architectural theory for environmental design research. In: Zube, E.H., Moore, G.T. (eds.) *Advances in Environment, Behavior, and Design*, pp. 3–52. Springer, Boston (1991). https://doi.org/10.1007/978-1-4684-5814-5_1
35. Roberts, L.: The Arpanet and computer networks. In: *A History of Personal Workstations*, pp. 141–172 (1988)
36. Mukand, S., Rodrik, D.: The political economy of ideas: on ideas versus interests in policymaking (no. w24467). National Bureau of Economic Research (2018)



Smart Helmet: An Experimental Helmet Security Add-On

David Sales¹, Paula Prata^{1,2,3}, and Paulo Fazendeiro^{1,2,3(✉)}

¹ Departamento de Informática (UBI-DI), Universidade da Beira Interior, Covilhã, Portugal

{david.sales,paulof}@ubi.pt, pprata@di.ubi.pt

² Instituto de Telecomunicações (IT), Universidade da Beira Interior, Covilhã, Portugal

³ C4 - Centro de Competências em Cloud Computing (C4-UBI), Universidade da Beira Interior, Covilhã, Portugal

Abstract. When it comes to ride a motorcycle the drivers-centered road safety is quintessential; every year a remarkable number of accidents directly related to sleepiness and fatigue occur. With the objective of maximizing the security on a motorcycle, the reported system aims to prevent sleepiness related accidents and to attenuate the effects of a crash. The system was developed as the less intrusive as it could be, with sensors that allow the capture of reaction times to stimuli-response and collect acceleration values. To obviate the lack of data related to sleepiness during motorcycle riding, a machine learning system was developed, based on Artificial Immune Systems. This way, resourcing to a minimum amount of user input, a custom system is synthesized for each user, allowing to assess the sleepiness level of each subject differently.

Keywords: Smart Helmet · Road safety · Motorcycle · Sleepiness · Machine learning · Reaction time · Artificial immune systems

1 Introduction

With the ever-growing safety solutions that are, currently, either vehicle embedded, or mere phone applications, it is easy to find systems that can predict and/or detect accidents. However the vast majority of these are most likely found in automobiles, and not in motorcycles. The system proposed in this work, *Smart Helmet*, was conceptualized and created with the intent of maximizing road safety for motorcyclists. The problems associated with road safety that are tackled by this system were: the driver's sleepiness and the aftermath of a motorcycle accident.

According to the data available on the National Sleep Foundation's website [1], from the 2005 Sleep in America poll, about 60% of the adults have confessed that they have driven while experiencing sleepiness (this corresponds to around 168 million people). The fact that sleepiness is a common road safety issue is

also emphasized by the National Highway Traffic Safety Administration (USA), having recorded 4111 fatalities (between 2013 and 2017), 795 deaths (only in 2017) and 91000 accidents (in 2017) all sleepiness-related [2].

In Portugal, the Road Safety National Authority (*Autoridade Nacional de Segurança Rodoviária*), in its 2018 annual reports of victims in road accidents within 24 h and 30 d of the accident [3], reveals that of the 34235 accidents present in both reports, there were 508 deaths (24 h report) and 675 fatalities (30 d report) in accidents caused by sleepiness.

Considering the above figures, we can say that there is a noticeable amount of people who get affected by the effects of sleepiness while driving, and also a considerable amount of accidents in motorcycles. Taking in mind these two issues, there is a need for a motorcycle safety system with the following features:

- Capability to detect the driver's sleepiness, through some techniques that will be discussed later on, with notifications associated, to prevent possible accidents.
- Identification of crashes using an accelerometer built-in into the system (placed on the helmet). After the correct analysis of a crash, the system will notify the selected emergency contacts with the current location, aiming at quickly aiding the user and possibly reducing any crash related injury.

Because one of the issues is related to the driver's attention to the road, *Smart Helmet* was developed in a way that is as non-intrusive as it can be. This way, the system can be added in a daily setting to the helmet without conflicting with the motorcyclists and their driving. The purpose of this system is to prevent crashes, as well as adding safety measures after a crash, while being available to the whole spectrum of motorcyclists.

Looking at the current market, at least the consumer grade one, there isn't any product that can really provide the same sense of security that this system can. There are some "Smart" helmets, in the sense that they can satisfy the multimedia needs of the motorcycle user, as well as voice control, GPS navigation, and extra vision by capturing the rear view (examples include *CrossHelmet* [4] and *Argon* [5]). There are phone applications that provide that bit of security, after a crash occurs, like *Realrider* [6], that after detecting a motorcycle crash send an SOS signal with your location.

The fatigue feeling that one has experienced while driving is not always sleepiness. A number of studies have been made around this matter (sleepiness) and, of those, many contemplate this feeling of sleepiness while driving. There are many ways of assessing this condition, such as: using an Electrocardiogram (or *ECG*) incorporated in the steering wheel of an automobile (as proposed in [7]), or measuring the sleepiness with the help of an Electroencephalogram (or *EEG*, used in an helmet in [8]).

Our proposed solution *Smart Helmet* combines the findings of the research done in the sleepiness field (mostly regarding driving) with the simplicity and availability of the commercial grade helmet add-ons. It maintains a minimum direct interaction with the motorcyclists while assessing their sleepiness level, preventing an important cause of casualties.

The remaining of this paper is structured as follows: in Sect. 2, some background knowledge regarding the techniques behind sleepiness assessment is given. Section 3 presents the final developed system, explaining its features, components, the *Artificial Immune Systems* approach to sleepiness detection and the implemented algorithms. A selected set of experiments and respective results, as well as the decisions taken along the development stage are presented in Sect. 4. Section 5 regards the conclusion and future work/experiments that can be done in the scope (and out as well) of this paper.

2 Background

In the context of driving, sleepiness is closely related with the propensity to stop paying attention to the road and progressively being overtaken by sleep. There are four methods that can be considered to “measure” sleepiness (according to [9]):

- Subjective Methods: Based on questionnaires made to the individuals in question. A scale is presented to them and they self-evaluate their sleepiness level. For example, the most common scale is the *Karolinska Sleepiness Scale* reproduced in Table 1, below. This scale presents nine levels of sleepiness from lowest (extremely alert) to highest (very sleepy, great effort to keep alert, fighting sleep), and this one is going to be used as reference in this paper further on. Although effective, it is not practical in this matter if not used together with another method.
- Vehicle-based Methods: Various sensors are put on various the parts of the vehicle that have a say in driving (example: brakes, gas pedal, . . .), these are going to give a correlation between the user’s tiredness and its behaviour with the vehicle. For example: braking too soon (to compensate their sleepiness) or too late. These aren’t too practical because of the intrusiveness in the motorcycle.
- Behavioral Methods: Based on the driver’s change in behavior, like, for example, yawning or the eye movements. One’s behavior is captured through a camera, and, because of that, can’t be considered for this system, although it has great success in automobiles.
- Physiological Methods: Through various factors of an individual (examples include: heartbeat, electrical activity of the brain, muscle activity), these measures are so accurate that it is possible to detect sleepiness in it’s early stages. The problem with this approach is its level of intrusiveness, every sensor has some level of it and, for that reason this method has to be discouraged.

Physiological measures are intrusive but accurate, according to [8], there is some correlation between the values obtained from the *EEG* and the subject’s reaction time. Moreover, [11] explores the quantification of sleepiness with and without *EEG* stating that through reaction times it is possible to evaluate the subject’s sleepiness. However, the test has to be simple enough so that the driver

Table 1. The karolinska sleepiness scale (*KSS*) as referred in [15].

Level	Verbal description
1	Extremely alert
2	Very alert
3	Alert
4	Fairly alert
5	Neither alert nor sleepy
6	Some signs of sleepiness
7	Sleepy, but no effort to keep awake
8	Sleepy, but some effort to keep alert
9	Very Sleepy, great effort to keep alert, fighting sleep

can do it without disrupting their driving and, the gathering of EEG signals from a helmet is cumbersome and somewhat ineffective (cf. [10]).

In [12], the drivers' response to danger stimuli after sleep deprivation is studied, and the conclusion is that the reaction time increases with sleep deprivation, as well as the *KSS* level. Another work [13], that focuses on the sleep restriction and its correlation with attention and reaction time, concludes that reaction time is prolonged with sleep restriction.

The work presented in [14], explores the correlation between a subjects' sleepiness and reaction time by doing an auditory test during 10 min before and after sleep deprivation. During those 10 min, various auditory stimuli are produced, and each subject has to press a button when they hear it, as fast as they can. There was a correlation between the two components, the reaction time increased. With everything said so far, it is possible to, while taking in mind the correlation between *KSS* levels and reaction time, make a specific 10-minute test adapted to the motorcycle driving setting, during that time the average reaction time and standard deviation are considered, and classified as a *KSS* level.

3 Developed System

The data that would be available to the system in question would be the accelerometer values for each axis (x, y, z), extracted from a device latched to an helmet. This, along with a microcontroller/computer with moderate processing power, is enough. As an obvious initial approach, machine learning techniques could be applied to conquer the objectives of this system. Because of the initial lack of historical accelerometer data, a user based approach was used, by letting the user interact with the agent, it will give an opportunity for, said agent, to grow its "knowledge" progressively. And so, an *Artificial Immune System (AIS)* [16] was used to represent each situation of the system. Regarding the implemented AIS *self-nonsel discrimination* is a core key concept, that is the ability to distinguish what is part of the immune system or not, being *self* the

system and non-self everything that isn't. When defining an immune system, self-nonself discrimination techniques are used to identify threats to the system. For example: an unknown cell can be harmful to a system depending on what the system perceives being part of it or not. A clear example can be observed between the two yellow cells in the Fig. 1.

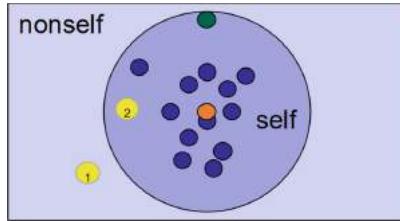


Fig. 1. Example of the representation of the crash detection system in R^2 . the blue circle represents the hypersphere. the green cell is the furthest cell from the centroid, the orange cell. yellow cells represent 2 different scenarios.

By using such approach, a model of each situation is created per each individual, and this way an agent is specifically catered to each user's way of interacting with the environment. *AIS*'s are a way of representing systems, inspired by the immune systems of vertebrates, in a way that, in an euclidean space, a representation of a specific system is made, in which said system has cells with certain characteristics specific to the system in question. Figure 2 presents the final system's architecture.

The choice of the components that are depicted in Fig. 2) was as follows:

- Raspberry Pi 3 B+ (*RPI*): This small computer possesses enough processing power to engage in such problem. Because of its capabilities of networking and connectivity, bluetooth connections can be made to communicate with other devices, in this system an *Android* phone is considered. The whole *AIS* approach was implemented here, using *Python* programming language. Disadvantages associated with the *RPI* are the fact that it still is a rather large component to attach to an helmet, some adaptations have to be made in the future.
- Buzzer and Button: To produce simple auditory stimuli, to gather the reaction times and evaluate one's sleepiness level, a simple buzzer was added. Also, a button serves simple I/O functionalities, that will be expressed later on (regarding the progress of a motorcycle ride).
- Sensors: The only sensor used was an accelerometer (LIS3DH [17] specifically, because it can reach 16G values, which are enough for the scope of the work), this because all the values gathered from it can be used to represent a normal situation and an accident situation (will be explained further on), as well as representing the stimulus answers, that will compose a sleepiness level. If head injury detection, or impact analysis, were to be added to this work, a

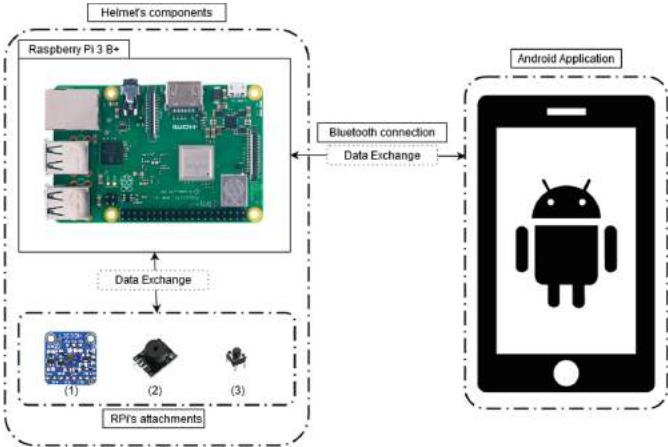


Fig. 2. System architecture, regarding the whole “Smart Helmet” environment. where the numbers refer to the *RPI*’s attachments, and they are, respectively: accelerometer, buzzer and button.

higher shock accelerometer would have to be used (much higher than 16G, in order to evaluate values that can be harmful to human beings).

- *Android Application:* Because most of the users possess an android phone, and the user interface on the *RPI* isn’t as friendly, an application was developed with the intent of communicating with the part of the system latched to the helmet. This will let the user interact with the *AIS* (*RPI*) through a *bluetooth* connection. The user can train the agent and perform rides (this means they can start the agent and set it on the lookout for possible accidents, as well as detecting the user’s sleepiness level from time to time).

3.1 Mode of Operation

The “Smart Helmet” depends on its associated Android application (*app*), to turning on the helmet-side system and start the system utilization. In the first time it is launched, a small trial (of 15 min) has to be done to gather common information to build the two models (crash detection and sleepiness evaluation). Before starting a trial, a subjective sleepiness evaluation has to be done within the *KSS*, table 1. During a trial, accelerometer data will be retrieved (as **acceleration norm**, that translates to $\sqrt{x^2 + y^2 + z^2}$, being x, y and z each acceleration component), and used for two models: crash detection (data representing the whole 15 min) and sleepiness detection (10 min, starting after the first 5, along with the *KSS* level given at the beginning).

In order to use the modeled systems, one has to begin a motorcycle ride after pressing the respective button (in the app), and, before actually starting, the current sleepiness has to be self-evaluated. This because the system needs to know how the user is feeling initially, and the amount of data available to evaluate

the user's sleepiness level can be low (and initially it will be pretty low to make a correct assumption). During a ride, the system will gather accelerometer data (every second) and try to find out if it represents a situation out of the ordinary or not, if so, an accident might have been detected (the word *might* is used, in this context, because various factors like bumps in the road, and so on, have to be considered). On crash detected, the user can respond if the accident has actually happened (on the helmet, through the button). One click and it uses that data to re-model the crash detection system (not a crash), two clicks and it asks for help (a crash happened, an SOS message is sent, along with the location, to the user's selected emergency contacts, that can be set in the application). Still regarding the detection of a crash, if the user doesn't respond in 30 s, an automatic SOS message is sent to the user's emergency contacts. The user can also answer in the application, in case of a crash detection and, if the user whether wants to send an SOS message or not.

Along a ride, the sleepiness is going to be evaluated, in case it is mild or none (below, and including level 4 of the KSS) nothing happens, between levels 5 and 6 an auditory signal is played that reminds the user they may want to rest for a little bit, and if 7 or above a signal is played that advises the user to stop and rest for 30 min, or take a nap/sleep. During a ride not all levels of sleepiness will be correctly evaluated (during the initial states of the system), this means that, because there is lack of initial user data, and the modeled sleepiness system can't find a way of classifying **all** of the incoming data (levels that aren't yet classified), it will try to give a response based on it's current system state. But, the response given will always be according to the worst case scenario (taking in mind the current system state, and levels evaluated during the ride). This way the motorcyclist will never be undermined by the sleepiness evaluation. Needless to say that these evaluations wont be fully trustable, and so, at the end of a ride the user will have to answer a questionnaire regarding these evaluations. While answering, the user can observe where and when the sleepiness level assessment took place, and the answer will have to be either true or false. This way, all the levels that aren't fully trustable, but are correct, can be used to re-model the sleepiness evaluation system, allowing it to evaluate a wider range of sleepiness levels. As one can see, the system adapts along the time, with fewer and fewer user interventions. The prototype of the developed system (helmet part) can be observed in Fig. 3.

3.2 Crash Detection

This immune system is composed of various cells that are part the definition of the self, viz. a set representing the normal state of operation. The characteristics of each cell are: average acceleration norm, maximum and minimum acceleration norm and the standard deviation. Each cell is the representation of one second worth of data. For example, in a trial, for each second of a 15 min interval a cell is going to be created and, in the end, they are going to be consolidated. That final consolidation is what the model is going to look like, so in this final part, the system finds the central cell (or centroid) of the agglomerate of cells,



Fig. 3. Prototype of system developed, the *RPI* is underneath the tape (at the side of the Helmet). a power bank can also be observed (it is used to power the *RPI*).

and computes the distance from the centroid towards the cell that is furthest apart. With a center on the agglomerate and a maximum distance defined, a hypersphere is created (because each cell has four characteristics, the euclidean space is R^4). This hypersphere then defines what a normal situation is, and to get that distinction, an input cell is taken and, only if that cell presents itself inside the defined field (hypersphere), that cell represents a normal situation, and not, for example, a crash. In Fig. 1 a simplified representation of the crash detection system can be observed. In reality the system is in R^4 , because cells have four characteristics, in the figure cell number 1 is an example of, let's say, a crash, and, cell number 2 a regular situation, using self-nonself discrimination.

3.3 Sleepiness and Fatigue Evaluation

In this case was chosen a different representation from the crash detection one because, since there are various *KSS* levels, each level has its own immune system. Before further explanation, all levels are going to be similar systems, apart from the position of the cells and the level they represent. And the cells that compose each level, called *sleepiness cells*, represent one 10 min reaction time test, that is composed by various reactions to stimuli, called *reaction cells*. So, every 10 min test, various *reaction cells* are extracted, these represent the processed data before the stimulus until a short time after the stimulus, this processed data is the reaction time it took for the user to answer a stimulus, the data processing is going to be introduced later on. A sleepiness cell takes all of the reaction cells, that are essentially reaction times for the stimuli in the 10 min test, and defines itself by the following characteristics: average reaction time and standard deviation. Each level is going to be composed by an agglomerate of sleepiness cells, although they are represented separately, they are all in the same R^2 euclidean space. The agglomerate that represents each level is a representation of the immune system of each level, when building a level the average distance between every cell (or in other words, the **threshold**), within a certain

level, is going to be computed and used to define the area of each sleepiness cell that composes a certain KSS level. So, a level is defined by the area of its sleepiness cells, and, using self-nonself discrimination, a cell belongs to a certain level if it is within the area of a level. Other techniques to classify unknown cells are going to be discussed further on. In Fig. 4, there are three levels illustrated along with two yellow unknown cells, which one of them belongs to level 3 and the other is going to undergo the techniques expressed further on, to classify it.

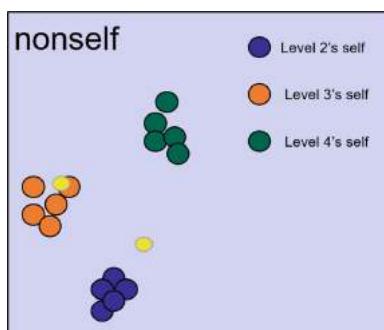


Fig. 4. Illustration of what the sleepiness model is, each agglomerate corresponds to a level, each circle is a cell, and yellow cells are unknown cells without any particular size.

One of the problems associated with the sleepiness evaluation system is the fact that, data extracted from the real world signifies such small bits of data in the system itself, for example if four 10-min reaction time tests are done, a total of about 40 or 50 reactions are extracted but those are represented in a total of only four sleepiness cells, that can even only represent one level. Through some algorithms that predict that classification of a level, this problem can be resolved. Side note, the crash detection system doesn't suffer from this because it ranks its cells in a binary classification as representing self or non-self. Because the reaction times regarding a certain *KSS* level can be very specific, the classification of a cell has to be very precise, and can never be predicted as a level that can be dangerously wrong, in the worst case scenario. So, later on, if the classification has to lean on the last stages, the worst case scenario (within the information available) is going to be predicted, meaning a higher level on the scale might be predicted instead of the current level. And so, to classify the level of an unknown cell, it undergoes a maximum of three stages, the first will try to classify it and the last two will try to give an approximation (it is important to state that predicted levels are levels that might not exist, were predicted through approximations, meaning that they might not still be modeled):

- The first stage is based on the k-NN algorithm, where the premise of the closest neighbors is changed a little bit, tailoring the needs of this system. Every cell of every level is going to be analyzed, and so, for each level, the

distance between the unknown cell and each cell is going to be calculated, and based on a limit distance defined for each level (called plausible distance, that is 2 times the threshold of a level), a ratio (which corresponds to the division of the distance, between the unknown cell and the current cell analyzed, by the plausible distance of the current level analyzed) is calculated between each cell and the unknown cell. If the ratio is below or equal to 1, regarding each cell of each level, that ratio along with the level (tuple) is added to a list, else it is disregarded. Then, the closest level (with the highest number of cells in the list) is verified and if there are no ties the level is returned, else a random level is picked within the list of closest levels (using the random integer generator from the module *random* of Python). If there aren't any close levels (the list is empty), the unknown cell fails this stage. For example: in Fig. 4 one can see that the top unknown cell would fit in this stage and be classified as part of level 3, and the bottom unknown cell would fail this prediction stage.

- The second stage will try to predict on what level the unknown cell might be included or what level it could represent, based on how much the reaction time of the unknown cell has increased, in relation to the average reaction time of the current level. It will be executed if the unknown cell presents itself above the last evaluated sleepiness level (current), and if there are any higher levels (modeled in the system) than the one evaluated before. Otherwise it will proceed to the next, and last, stage. The algorithm finds out the growth ratio between the reaction time of the unknown cell and the average reaction time of the current sleepiness level. With that, it verifies if the cell might be enclosed by two levels, or if it is above any known level that is equal or higher to the current. And, according to its growth, it verifies to which level it might be closer to, or what it could classify.
- The last stage really focuses on over-estimating the sleepiness level of the user, the growth ratio is again introduced. If the ratio between the unknown input cell and the current average reaction is below 20%, it is considered that the level has maintained the same, else it has probably risen. In case there weren't predicted any levels before (through any of these stages), this stage returns the current level plus one, else it will compare with the last predicted level (that isn't modeled). If the ratio (in relation to the reaction time of the last unknown cell, used to predict the previous level) is higher than 20% then the returned level is one level higher than the previous predicted, otherwise it stays the same level (previous predicted).

4 Selected Experiments and Results

The reported experiment had the sole intent of finding what patterns the acceleration values followed, after the motorcyclist responded to a stimulus. The protocol consisted on the extraction of the accelerometer data, over an extended period of time, and, every 30 to 60 s an auditory stimulus would be produced, to which the user was told to respond to in a comfortable manner (a simple

nod). It is valuable to add that the experiment also had an effort to maximize the quantity of data received, and also to be as close as possible to a real case scenario.

The *Savitzky-Golay* filter (more specifically, the function `signal.savgol_filter` from *SciPy* [18]) was used on all of the retrieved data, in order to turn rather noisy data smooth (so it could be more easily interpreted). Noisy data is inevitable, because of the constant vibration of the motorcycle's high-rotation engine. The parameters for said filter were obtained empirically, always considering the factors involved, and in a way that it didn't filter out too much data. A real example of such filtering can be observed in Fig. 5.

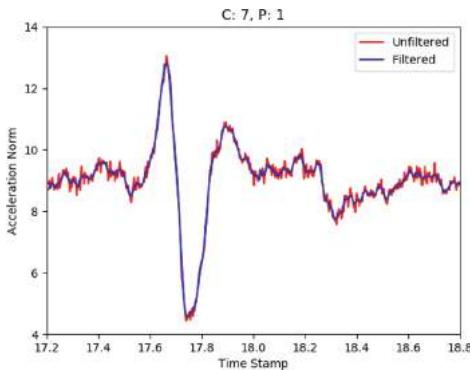


Fig. 5. The plotted red line is the raw accelerometer data unfiltered, the blue is the raw data filtered. the filter used was the *SciPy* implementation of *Savitzky-Golay* filter. the parameters were coefficient = 7 and polyorder = 1.

For the matter of a real case scenario, the subject was a motorcyclist that rode their own motorcycle through a common route, for a minimum of 15 min. Figure 6 illustrates the first 200 s of data recorded for a particular ride.

In the figure the yellow lines express the exact moment in that an auditory stimulus was produced, the plotted blue line represents the acceleration norm (y-axis) over time (seconds, x-axis), the red rectangles are two arbitrary cases used for analysis later in this report (that will concern how a reaction cell is obtained from raw data) and, the green lines are the threshold to which the nod of the user should surpass to be counted as a reaction to the stimulus.

In order to facilitate the analysis, on the left side of the Fig. 7 we can observe a magnification of the previously marked regions of Fig. 6. A first obvious pattern can be observed: after every auditory stimulus there is a peak in the accelerometer data. Moreover, after a stimulus the distribution of the accelerometer values appear to follow an approximate normal distribution as is depicted on the right side of the figure.

In the time-lapse prior to any stimulus, the distribution was calculated and, after said stimulus, the threshold was defined (to be able to identify most of the

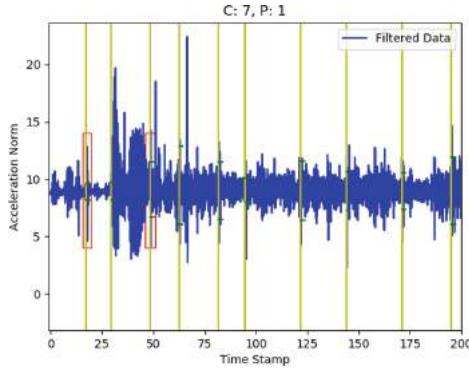


Fig. 6. Plotted acceleration values, already filtered, that concern the experiments exposed in this section. the “C” and “P” at the top mean *coefficient* and *polyorder*, respectively.

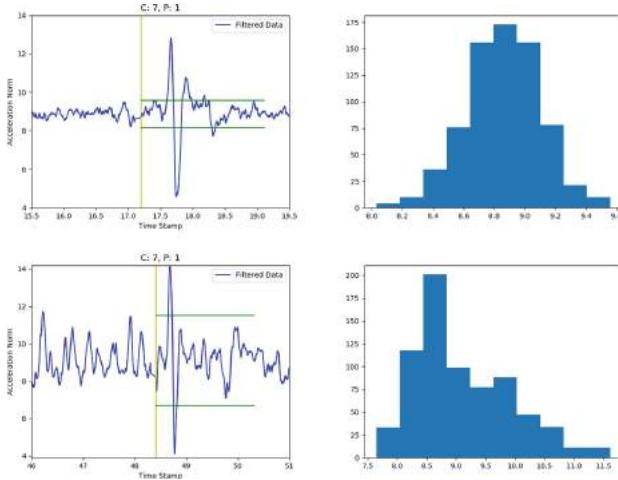


Fig. 7. This figure regards auditory stimuli highlighted in 6, respectively. at the left side the data is plotted, yellow lines represent the exact moment of a stimulus and green lines depict the threshold. at the right side the distribution of the values, before the stimulus was produced, can be found, for each case.

normal distribution values, about 99.7% of them, it was set to $\text{average} \pm 3 * \sigma$ of the prior distribution). If the threshold was surpassed (at the top, or bottom), it means that the system received abnormal data, considered to be a nod from the user. From this, a *reaction cell* was created, to later be a part of the conglomerate that constitutes a *sleepiness cell*. In the reaction cell construction, all of the data (about 4 s, 2 before and 2 after) was analyzed. From the first 2 s the distribution of the data was calculated so that, in the last 2 s, the threshold could be defined and, the reaction time was measured from the moment of the stimulus until the

threshold is firstly surpassed. Every time an auditory stimulus is produced, a reaction cell will be created and, it will be only used if there was identified an user response.

5 Conclusion

Getting inspiration from various studies surrounding the subject of road safety and sleepiness/drowsiness, a smart system was developed to improve the motorcyclists' road safety. It is important to note that additional findings and studies are much in need, namely, in the sleepiness field.

In our endeavour of building a non-intrusive easy to use system, we have adopted an indirect measurement of sleepiness depending on the speed of response to stimuli instead of any of the proven direct physiological methods. That is a recognized trade-off, since as previously noted in other studies (e.g. [19]) a system providing more *stimulation* for the sleepy driver, breaks monotony and can significantly reduce the subjective sleepiness, with a trend for fewer incidents.

On the proposed solution, *Artificial Immune Systems* were used to account for the little data available for both crash and sleepiness detection systems. A different set of machine learning techniques can be applied to arrange a more accurate and effective classification system, once the amount of collected data is gathered. Nonetheless, a good solution was proposed, the initial use has quite a bit of user interaction, but, progressively, the system learns the task of evaluating the user's sleepiness and driving styles. This way every user has a specific system, built right for them.

5.1 Future Work

Other alternatives to solve the challenges of this work include (i) the need to rethink design-wise the encasing method in order to attach the control unit to the helmet in an appealing way and, (ii) the assessment of direct physiological sleepiness evaluation methods as putative replacements for the reaction time approach.

If this system is introduced to the market a cloud-oriented implementation can arise and with its growth a lot of data can be gathered hence contributing to further studies in both sleepiness and motorcycle crash prevention. This means that, through an implementation based on the Cloud, not only this data can be used outside of the scope of this project, but it can also be used to compute new models for each user (based on their personal characteristics, and resourcing to other machine learning techniques), in order to research and find out if there is a better way of classifying incoming data, during a motorcycle ride.

Because of the scalability and flexibility of cloud computing, various applications could be developed in order to share data to further increase knowledge on this matter and, with it, comes a great and feasible opportunity of creating a new way of detecting crashes as well as, and more important, preventing them.

Acknowledgments. This work was supported by operation Centro-01-0145-FEDER-000019 - C4 - Centro de Competências em Cloud Computing, co-financed by the European Regional Development Fund (ERDF) through the Programa Operacional Regional do Centro (Centro 2020), in the scope of the Sistema de Apoio à Investigação Científica e Tecnológica - Programas Integrados de IC&DT. This work was also funded by FCT/MCTES through the project UIDB/50008/2020.

References

1. Facts and Stats - Drowsy Driving - Stay Alert, Arrive Alive <https://drowsydriving.org/about/facts-and-stats/>
2. Drowsy Driving—National Highway Traffic Safety Administration <https://www.nhtsa.gov/risky-driving/drowsy-driving>
3. Autoridade Nacional de Segurança Rodoviária - Relatórios de Sinistralidade <http://www.anvr.pt/Estatísticas/RelatóriosDeSinistralidade/Pages/default.aspx>
4. CrossHelmet X1 features: HUD, Bluetooth & much more <https://www.crosshelmet.com/features/index.html>
5. Argon Transform—Smart Augmented Reality Riding <https://www.argontransform.com/>
6. REALRIDER - The Motorcycle App - Safety - Routes <http://www.realrider.com/>
7. Lee, B.G., Chung, W.Y.: A smartphone-based driver safety monitoring system using data fusion. Sensors (Basel) **12**(12), 17536–17552 (2012). <https://doi.org/10.3390/s121217536>
8. Foong, R., Ang, K.K., Quek, C., Guan, C., Wai, A.A.P.: An analysis on driver drowsiness based on reaction time and EEG band power. In: Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) 2015, pp. 7982–7985 (2015). <https://doi.org/10.1109/EMBC.2015.7320244>
9. Sahayadhas, A., Sundaraj, K., Murugappan, M.: Detecting driver drowsiness based on sensors: a review. Sensors (Basel) **12**(12), 16937–16953 (2012). <https://doi.org/10.3390/s121216937>
10. Sun, Y., Yu, X.B.: An innovative nonintrusive driver assistance system for vital signal monitoring. IEEE J. Biomed. Health Inform. **18**(6), 1932–1939 (2014). <https://doi.org/10.1109/JBHI.2014.2305403>
11. Penzel, T., Fietze, I., Schöbel, C., Veauthier, C.: Technology to detect driver sleepiness. Sleep Med. Clin. **14**(4), 463–468 (2019). <https://doi.org/10.1016/j.jsmc.2019.08.004>
12. Mahajan, K., Velaga, N.R.: Effects of partial sleep deprivation on braking response of drivers in hazard scenarios. Accid. Anal. Prev. **142**, 105545 (2020). <https://doi.org/10.1016/j.aap.2020.105545>
13. Choudhary, A.K., Kishanrao, S.S., Dadaraao Dhanvijay, A.K., Alam, T.: Sleep restriction may lead to disruption in physiological attention and reaction time. Sleep Sci. **9**(3), 207–211 (2016). <https://doi.org/10.1016/j.slsci.2016.09.001>
14. Lisper, H.O., Kjellberg, A.: Effects of 24-hour sleep deprivation on rate of decrement in a 10-minute auditory reaction time task. J. Exp. Psychol. **96**(2), 287–290 (1972). <https://doi.org/10.1037/h0033615>
15. Sahayadhas, A., Sundaraj, K., Murugappan, M.: Drowsiness detection during different times of day using multiple features. Australas. Phys. Eng. Sci. Med. **36**, 243–250 (2013). <https://doi.org/10.1007/s13246-013-0200-6>. supported by the Australasian College of Physical Scientists in Medicine and the Australasian Association of Physical Sciences in Medicine

16. Fernandes, D.A., Freire, M.M., Fazendeiro, P.A., Inácio, P.R.: Applications of artificial immune systems to computer security: a survey. *J. Inf. Secur. Appl.* **35**, 138–159 (2017). <http://hdl.handle.net/10400.6/8203>, <https://doi.org/10.1016/j.jisa.2017.06.007>
17. Adafruit LIS3DH Triple-Axis Accelerometer Breakout <https://learn.adafruit.com/adafruit-lis3dh-triple-axis-accelerometer-breakout/overview>
18. SciPy—SciPy v1.5.2 Reference Guide <https://docs.scipy.org/doc/scipy/reference/index.html>
19. Baulk, S.D., Reyner, L.A., Horne, J.A.: Driver sleepiness—evaluation of reaction time measurement as a secondary task. *Sleep* **24**(6), 695–698 (2001)

Author Index

A

- Abadi, Hossein Karkeh, 754
Abdelhameed, Wael A., 1115
Abeywardena, Kavinga, 793
Abubahia, Ahmed, 524
Agrawal, Ayush Manish, 678
Agwu, Nwojo, 1056
Alelyani, Salem, 1151
Alihodzic, Adis, 341
Aliyev, Elchin, 13
AlKhawaldeh, Fatima T., 661
Al-Mubaid, Hisham, 261
Alonso, Luis, 940
Alphinas, R. A., 463
Al-Saadi, Bushra, 357
Al-Saadi, Muna, 357
Alsaif, Hasan, 80
Alsaqer, Mohammed Saleh, 1151
Anagnostopoulos, Christos, 619
Anzilli, Luca, 210
Apaza-Alanoca, Honorio, 492
Araújo, Luiz, 1169
Arciniega-Rocha, Ricardo P., 1073
Ayala, Orlando, 1222

B

- Bader-El-Den, Mohamed, 524
Baimukhamedov, Malik, 39
Bandara, Eranga, 891
Bankher, Ahmed, 1085
Barros, Vladimir, 1169
Bel Hadj Youssef, Soumaya, 825
Beling, Peter A., 912
Beloff, Natalia, 858

- Bhattacharya, Rituparna, 858
Black, Paul E., 881
Blazsik, Zoltan, 976
Bleile, MaryLena, 196
Borambayev, Askar, 39
Borambayev, Seilkhan, 39
Borbély, Tamás, 453
Boudriga, Noureddine, 825
Bunyard, Sara, 115
Burnaev, E., 591

C

- Camargo, Darcy, 927
Capossele, Angelo, 840
Castaldi, Paolo, 23
Chen, Haonan, 232
Chepkov, Roman, 97
Cho, Michael Cheng Yi, 65
Contreras, M. Leonor, 371
Crocket, Keeley, 49
Cruz, John Emmanuel B., 1124
Cui, Yuanjun, 743
Cunjalo, Fikret, 341

D

- D’Souza, Daryl, 174
da Silva, Isabela Ruiz Roque, 869
Das Gupta, Dipannoy, 295
Datta, Siddhartha, 637
De Zoysa, Kasun, 891
DeLise, Timothy, 139
Do, Thuat, 805
Duan, Paul, 743
Dzwonkowski, Adam, 284

E

- Ebrahimi, Mohammad Sadegh, 754
 Elek, Istvan, 976
 Elkatsha, Markus, 940
 El-Sharkawy, Mohamed, 781
 Erdem, Atakan, 273
 Eryong, Li, 1141

F

- Fakieh, Bahjat, 1085
 Farrell, Steven, 473
 Farsoni, Saverio, 23
 Fazendeiro, Paulo, 1236
 Felemban, Emad, 1106
 Flores, Anibal, 492
 Fossi, Jose, 284
 Fotouhi, Farshad, 408
 Fouda Ndjodo, Marcel, 427
 Fountas, Panagiotis, 619
 Foytik, Peter, 891
 Fry, John, 49

G

- Garcés-Báez, Alfonso, 952
 Gerber, Luciano, 49
 Gibson, Ryan M., 1032
 Gil, Santiago, 1010
 Giove, Silvio, 210
 Gregorics, Tibor, 453
 Grignard, Arnaud, 940
 Griguta, Vlad-Marius, 49
 Grooss, O. F., 463

H

- Haig, Ella, 524
 Haiko, Svitlana, 97
 Han, Yi, 743
 Hasan, Mahady, 581
 Hasanspahic, Damir, 341
 He, Jing, 232
 Heger, Tamas, 976
 Henson, Cory, 325
 Herman, Maya, 607
 Holm, C. N., 463
 Huang, Cayden, 743
 Huang, Hsiu-Chuan, 65
 Huang, Kevin, 325
 Hussain, Mohammad Rashid, 1151

I

- Ikechukwu, Onyenwe, 1056
 Ingabire, Winfred, 1032
 Iskandarani, Mahmoud Zaki, 1203
 Islam, Md. Saiful, 581

Islam, Tanhim, 581

Ivanovna, Sipovskaya Yana, 249

J

- Jabrah, Mohammad, 1085
 Jackson, Karen Moran, 725
 Jara-Figueroa, Cristian Ignacio, 940
 Jayasinghe, D. P. P., 793
 Jiang, Hui, 398
 Jiang, Shenfei, 743

K

- Kayid, Amr, 678
 Kazakov, Dimitar, 661
 Kelefouras, Vasilios, 357
 Khan, Asiya, 357
 Kim, Ji Eun, 325
 Kodati, Meenakshi, 129
 Kolomvatsos, Kostas, 507, 619
 Kurth, Thorsten, 473
 Kuśmierz, Bartosz, 840

L

- Laakso, Mikko-Jussi, 174
 Larijani, Hadi, 1032
 Larson, Kent, 940
 Lbath, Ahmed, 1106
 Lenger, Daniel, 976
 Levi, Ofer, 607
 Lghoul, Rachid, 1044
 Li, Xiaofeng, 648
 Li, ZhiQiang, 764
 Liang, Xueping, 891
 Licea Torres, Luis David, 261
 Lin, Siyu, 912
 Lin, Wan-Yi, 325
 Liyanage, Romesh, 793
 Llanes-Cedeño, Edilberto A., 1073
 López-López, Aurelio, 952
 López-Villada, Jesús, 1073
 Lu, Minggui, 743
 Luna, Sanzida Mojib, 295

M

- Macabebé, Erees Queen B., 1124
 Maguire, Phil, 152
 Majid, Abdur Rahman Muhammad Abdul, 1106
 Manna, Sukanya, 115
 Masmoudi, Ilias, 1044
 Matovu, Richard, 308
 Mazyavkina, N., 591
 Meng, Shawn, 743
 Miao, YaBo, 398
 Miller, Robert, 152
 Montesclaros, Ray Mart M., 1124

- Mousavi Mojab, Seyed Ziae, 408
Moustafa, S., 591
Mudrak, George, 1190
Müller, Sebastian, 840, 927
- N**
Nadutenko, Maksym, 97
Ndognkon Manga, Maxwell, 427
Ng, Wee Keong, 891
Nurbekov, Askar, 39
Nwokeji, Joshua C., 308
- O**
Obianuju, Nzurumike L., 1056
Oliveira, Eduardo, 1169
Omar, Nizam, 869
Osadchy, Volodymyr, 987
Othman, Salem, 80, 284
- P**
Pagoulatou, Tita, 562
Palmer, Xavier-Lewis, 1222
Pan, Yue, 743
Papa, Rosemary, 725
Parocha, Raymark C., 1124
Pearson, H. B. D. R., 793
Penzkofer, Andreas, 927
Pietroń, Marcin, 712
Pintér, Balázs, 453
Pirouz, Matin, 1
Popova, Maryna, 97
Potter, Lucas, 1222
Powell, Ernestine, 1222
Prata, Paula, 1236
Prykhodniuk, Vitalii, 97
- Q**
Qahmash, Ayman, 1151
Qu, ShaoJie, 764
- R**
Ranasinghe, Nalin, 891
Rehman, Faizan Ur, 1106
Roesch, Phil, 80
Romanyuk, Kirill, 221
Rosado, Bryan, 881
Rosero-Garcia, Jhonatan F., 1073
Rozas, Roberto, 371
Rzayev, Ramin, 13
- S**
Saa, Olivia, 927
Salakoski, Tapio, 174
Sales, David, 1236
Salmanov, Fuad, 13
- Sarwar, Hasan, 295
Sawarkar, Kunal, 129
Schmeelk, Suzanna, 881
Scola, Leila, 378
Semwal, Sudhanshu Kumar, 1190
Shah, Prasham, 781
Shams, Seyedmohammad, 408
Shetty, Sachin, 891
Sierra, Rodolfo García, 1010
Sikka, Harshvardhan, 678, 967
Sikka, Sidhdharth, 967
Simani, Silvio, 23
Sindely, Daniel, 976
Singh, Sahib, 678
Skuratovskii, Ruslan V., 987
Slater-Petty, Helen, 49
Smajlovic, Haris, 341
Soltanian-Zadeh, Hamid, 408
Souza, Caio, 694
Sriyaratna, Disni, 793
Stamoulis, George, 507
Stryzhak, Oleksandr, 97
Su, Yuan-Hsiang, 65
Suhi, Nusrat Jahan, 295
Sun, Shu, 648
- T**
Tam, Eugene, 743
Tasnim, Marzouka, 295
Tendle, Atharva, 678
Tito-Chura, Hugo, 492
Tran, Tuan A., 325
Trofimov, I., 591
Tsai, Yu-Lung, 65
Tsanakas, Stylianos, 562
Tserpes, Konstantinos, 562
- U**
Uvindu Sanjana, K. T., 793
- V**
Varvarigou, Theodora, 562
Veerasamy, Ashok Kumar, 174
Velhor, Luiz, 694
Villarroel, Ignacio, 371
Violos, John, 562
- W**
Walker, David J., 357
Wang, Chuyi, 232
Wang, Yunsong, 473
White, Martin, 858
Wiatr, Kazimierz, 712
Wild, Árpád János, 453
Williams, Samuel, 473
Wu, Shu-Fei, 166

X

- Xie, Jacke, [743](#)
Xu, FangYao, [764](#)

Y

- Yang, Charlene, [473](#)
Yu, Jinsong, [743](#)
Yuan, Tommy, [661](#)
Yue, JiaQi, [764](#)

Yukun, Li, [1141](#)

Yurrita, Mireia, [940](#)

Z

- Zagagy, Ben, [607](#)
Zapata-Madrigal, Germán D., [1010](#)
Zhang, Yan, [940](#)
Zhang, Ziqi, [543](#)
Zioviris, Georgios, [507](#)
Žurek, Dominik, [712](#)